



Universidad Internacional de La Rioja
Escuela Superior de Ingeniería y
Tecnología

Máster Universitario en Inteligencia Artificial
Sistema de recomendación de películas basado en análisis
emocional con *chatbot, Angella*

Trabajo fin de estudio presentado por:	Rubén Darío Aponte Núñez Edison José Giraldo Pérez Joel Elí Orellana Martínez
Tipo de trabajo:	Desarrollo Software
Director/a:	Rodrigo Gil-Merino y Rubio
Fecha:	20/09/2023

Resumen

El presente trabajo de fin de máster se enfoca en el diseño de un sistema de recomendación de películas basado en el análisis de las emociones, implementando un sistema de conversación interactivo llamado AngellA. Este tiene como principal objetivo optimizar la experiencia del usuario, proponiendo películas que sean acordes a sus estados emocionales predominantes. La metodología adoptada para este proyecto se estructura en distintas etapas: la recopilación de datos cinematográficos, el desarrollo y entrenamiento de un modelo especializado para el análisis de las emociones y la configuración y despliegue del sistema de recomendaciones. Los resultados muestran una capacidad notable de clasificar con precisión las emociones, lo que resulta en recomendaciones cinematográficas pertinentes. Por su parte, las evaluaciones ponen de manifiesto la habilidad del sistema para entregar sugerencias que estén acordes con el perfil emocional del usuario. Es evidente que la aplicación del análisis de las emociones en este contexto potencia de manera significativa la personalización de las recomendaciones, lo que a su vez conduce a un aumento en la satisfacción del usuario.

Palabras clave: análisis emocional, chatbot, películas, procesamiento del lenguaje natural, sistema de recomendación.

Abstract

This master's thesis focuses on the design of a movie recommendation system based on the analysis of emotions, implementing an interactive conversation system called AngellA. The main objective of this is to optimize the user experience, proposing films that are consistent with their predominant emotional states. The methodology adopted for this project is structured in different stages: the collection of cinematographic data, the development and training of a specialized model for the analysis of emotions, and the configuration and deployment of the recommendation system. The results show a remarkable ability to accurately classify emotions, resulting in relevant movie recommendations. Additionally, the evaluations reveal the system's ability to deliver suggestions that are in line with the user's emotional profile. It is evident that the application of emotion analysis in this context significantly enhances the personalization of recommendations, which in turn leads to an increase in user satisfaction.

Keywords: chatbot, emotional analysis, movies, natural language processing, recommendation system.

Índice de contenidos

1. Introducción	11
1.1. Motivación	12
1.2. Planteamiento del problema	14
1.3. Estructura de la memoria	14
2. Contexto y estado de la técnica	16
2.1. Sentimientos y emociones: definición	16
2.2. Neurociencia de los sentimientos y emociones	17
2.2.1. Procesamiento de las señales emocionales en el cerebro	18
2.2.2. Las Vías y Circuitos Emocionales	19
2.3. Avances en el estudio de emociones	19
2.4. Avances en la detección de sentimientos y emociones en texto mediante IA	22
2.5. Enfoques en la extracción de emociones	23
2.6. Desafíos en la detección de sentimientos y emociones	24
2.7. Sistemas de recomendación	25
2.7.1. Definición	25
2.7.2. Taxonomía de los sistemas de recomendación	26
2.7.3. Categorías de sistemas de recomendación	28
2.7.1. Modelado de un sistema de recomendación basado en NLP	30
2.8. <i>Chatbots</i> en sistemas de recomendación	32
2.8.1. Arquitectura de un sistema de chatbot	35
2.9. Desafíos y oportunidades en la investigación	35
3. Objetivos y metodología de trabajo	37
3.1. Objetivo general	37
3.2. Objetivos específicos	37

3.3.	Metodología del trabajo	37
4.	Identificación de requisitos	39
4.1.	Conjunto de datos de las películas	39
4.2.	Modelos de Deep Learning para la detección de emociones.....	41
4.3.	Traducción automática para inputs del usuario	42
4.4.	Frameworks para la creación de chatbots.....	42
4.5.	Características del flujo de conversación	44
4.6.	Interfaces de interacción para chatbots	45
4.7.	Computación en la nube	47
5.	Descripción de la herramienta software desarrollada.....	48
5.1.	Creación del conjunto de datos de películas	48
5.1.1.	Exploración de los diferentes conjuntos de datos de películas disponibles en la Internet 48	
5.1.2.	Preparación del conjunto de datos personalizado.....	51
5.2.	Creación del modelo para el análisis de emociones.....	56
5.2.1.	Entrenamiento del modelo.....	58
5.2.2.	Aplicación del modelo de clasificación al conjunto de datos de películas.....	68
5.2.3.	Segmentación del conjunto de datos de películas.....	71
5.2.4.	Reajuste emocional para recomendación de películas	75
5.3.	Desarrollo del sistema de recomendación y chatbot	78
5.3.1.	Descripción del sistema de recomendación propuesto	78
5.3.2.	Chatbot: AngellA.....	81
5.3.3.	Elección del sistema de generación de texto del chatbot.....	81
5.3.4.	Backend	83
5.3.5.	Frontend	86

5.3.6.	Registro del usuario en el chatbot AngellA	90
5.3.7.	Control de versiones del software.....	91
5.4.	Despliegue del sistema de chatbot: AngellA	92
5.4.1.	Configuración del servidor.....	92
5.4.2.	Configuración de dominio	94
5.4.3.	Creación de Dockerfile.....	94
5.4.4.	Construcción y ejecución de la imagen en Docker	96
6.	Evaluación del chatbot	97
6.1.	Pruebas funcionales del sistema de recomendaciones.....	97
6.1.1.	Evaluación de la influencia del sistema de traducción automática	97
6.1.2.	Prueba del sistema en un contexto emocional altamente negativo	99
6.1.3.	Prueba del sistema en un contexto emocional altamente positivo.....	101
6.2.	Pruebas realizadas por terceros	104
7.	Conclusiones y trabajos futuros	115
7.1.	Conclusiones	115
7.2.	Trabajos Futuros	116
	Referencias bibliográficas.....	118
	Anexo A. Artículo de investigación	128

Índice de figuras

Figura 1. Sistema límbico encargado de coordinar las respuestas emocionales.	18
Figura 2. Rueda de las emociones, donde se muestra su estructura y jerarquía.	20
Figura 3. Diagrama de flujo para sistema de recomendación usando NLP.....	32
Figura 4. Esquema básico de la arquitectura de un chatbot.....	36
Figura 5. Arquitectura MTV de Django.....	44
Figura 6. Estructura del sistema de archivos del conjunto de datos TMDb.....	52
Figura 7. Creación de dataframes por género de películas.....	53
Figura 8. Creación de todos los dataframes basado en las columnas detalladas con el parámetro “on”.	54
Figura 9. Combinación de las características principales de las películas con sus reseñas.	55
Figura 10. Conjunto de datos final con las características principales de las películas y sus reseñas.....	55
Figura 11. Modelo “distilbert-base-uncased” con fine tuning para clasificación de emociones.....	57
Figura 12. Tweets presentes en el conjunto de datos DAIR con sus emociones.	58
Figura 13. Importación del conjunto de datos para entrenamiento del modelo de clasificación de emociones.	59
Figura 14. Frecuencia de emociones presentes en el conjunto de datos de tweets.	60
Figura 15. Numero de palabras por tweet presentes en el conjunto de datos.	61
Figura 16. Tokenización del conjunto de datos de tweets.....	61
Figura 17. Muestra de embedding y máscara de atención para un tweet del conjunto de datos.	62
Figura 18. Entrenamiento del modelo clasificador de emociones.....	64
Figura 19. Proceso de entrenamiento del modelo clasificador de emociones.	64
Figura 20. Creación de la matriz de confusión para evaluar la precisión del modelo.	65
Figura 21. Matriz de confusión para evaluar la precisión del modelo de clasificación de emociones.....	66
Figura 22. Creación de modelo de clasificación de emociones en la plataforma Hugging Face.	67
Figura 23. Modelo de clasificación de emociones cargado en la plataforma Hugging Face....	67

Figura 24. Función para aplicación del modelo de clasificación de emociones al conjunto de datos de películas.	68
Figura 25. Aplicación del modelo de clasificación de emociones al conjunto de datos de películas.	69
Figura 26. Dataframe de películas con la columna de clasificación de emociones.	69
Figura 27. Función para aplicar promedio de emociones de las reseñas por película.	70
Figura 28. Dataframe con promedio de emociones de las reseñas por película.	70
Figura 29. Dataframe de películas con análisis de emociones y sinopsis.	70
Figura 30. Conjunto de datos codificado para ser usado en el análisis de segmentación.	72
Figura 31. Variación de la distorsión con el número de agrupaciones o clusters.	74
Figura 32. Cantidad de películas en cada agrupación o clúster.	74
Figura 33. Distribución de las diez agrupaciones usando dos componentes principales.	75
Figura 34. Contrapartes de emociones implementadas en el chatbot.	76
Figura 35. Función de balance de emociones implementadas en el chatbot.	77
Figura 36. Ejemplo de funcionamiento del reajuste emocional en el chatbot.	78
Figura 37. Diagrama de flujo del funcionamiento simplificado de AngellA.	82
Figura 38. Función analizadora de emociones del backend.	84
Figura 39. Función analizadora de oraciones del backend.	85
Figura 40. Función para cargar el conjunto de datos de películas del backend.	86
Figura 41. Función para filtrar el conjunto de datos por edades del backend.	86
Figura 42. Página web del chatbot AngellA.	87
Figura 43. Flujo conversacional del chatbot AngellA.	89
Figura 44. Recomendaciones de películas del chatbot AngellA.	89
Figura 45. Página de registro de usuario en el chatbot AngellA.	90
Figura 46. Términos y condiciones de uso del chatbot AngellA.	91
Figura 47. Repositorio GitHub del chatbot AngellA.	92
Figura 48. Arquitectura de red del servidor para el chatbot AngellA.	94
Figura 49. Archivo Dockerfile que contiene las instrucciones de configuración para despliegue en el servidor.	95
Figura 50. Construcción de la imagen en Docker para el chatbot AngellA.	96
Figura 51. Ejecución de la imagen en Docker para el chatbot AngellA.	97

Figura 52. Recomendaciones realizadas por AngellA a respuestas altamente negativas.

Izquierda: primeras cinco recomendaciones. Derecha: Segundas cinco recomendaciones. 100

Figura 53. Recomendaciones realizadas por AngellA a respuestas altamente positivas.

Izquierda: primeras cinco recomendaciones. Derecha: Segundas cinco recomendaciones. 102

Figura 54. Modelo entidad - relación en el administrador de Django para las calificaciones de los usuarios. 105

Figura 55. Número de películas por cantidad de recomendaciones..... 106

Figura 56. Conteo de calificaciones de las películas proporcionadas por los usuarios. 107

Figura 57. Análisis demográfico de los usuarios del chatbot. 108

Figura 58. Análisis de películas recomendadas por año de lanzamiento. 109

Figura 59. Nube de palabras que aparecen con mayor frecuencia en el chatbot. 110

Figura 60. Distribución de la cantidad de palabras en las respuestas de los usuarios. 111

Figura 61. Relación entre las calificaciones y la solicitud de nuevas recomendaciones hechas por los usuarios..... 112

Figura 62. Recomendación de película con emoción del usuario altamente positiva. 114

Figura 63. Recomendación de película con emociones mixtas del usuario. 114

Figura 64. Recomendación de película con emoción del usuario altamente negativa..... 115

Índice de tablas

Tabla 1. Diferentes emociones y sentimientos asociados a cada una de ellas.	20
Tabla 2. Modelos para las diferentes emociones básicas propuestos por diferentes autores.	21
Tabla 3. Comparativo de diferentes enfoques de sistemas de recomendación.	31
Tabla 4. Crecimiento de parámetros por año para los LLM.	34
Tabla 5. Parámetros de entrenamiento del modelo de clasificación de emociones.	62
Tabla 6. Resultado de métricas del modelo clasificador de emociones.	65
Tabla 7. Especificaciones de la máquina virtual de GCP.	93
Tabla 8. Preguntas y respuestas en español e inglés ingresadas al chatbot.	98
Tabla 9. Recomendaciones realizadas a las respuestas en español e inglés ingresadas al chatbot.....	98
Tabla 10. Respuestas en español altamente negativas ingresadas al chatbot.	99
Tabla 11. Primeras recomendaciones para respuestas en español altamente negativas.	100
Tabla 12. Segundas recomendaciones para respuestas en español altamente negativas. ...	101
Tabla 13. Respuestas en español altamente positivas ingresadas al chatbot.	102
Tabla 14. Primeras recomendaciones para respuestas en español altamente positivas.	103
Tabla 15. Segundas recomendaciones para respuestas en español altamente positivas.	103

1. Introducción

En la era actual de la información y el entretenimiento, las personas se enfrentan a una gran cantidad de opciones de películas y contenido multimedia, lo que hace que encontrar películas que se ajusten a sus preferencias y estados de ánimo particulares sea un desafío. Los sistemas de recomendación de películas han sido ampliamente investigados para abordar este problema (Adomavicius & Tuzhilin, 2005). Sin embargo, las técnicas tradicionales de recomendación consistente en el filtrado colaborativo pueden no ser suficientes para capturar las preferencias y emociones cambiantes de los usuarios en tiempo real (Su & Khoshgoftaar, 2009). Además, la interacción entre los usuarios y los sistemas de recomendación suele ser limitada y carece de una comunicación dinámica y adaptativa que permita refinar las recomendaciones en función de las necesidades específicas de cada usuario.

En este trabajo, se propone *un sistema de recomendación de películas basado en análisis emocional integrado en un chatbot* (Dahiya, 2017), llamado “*AngellA*”, en referencia a la palabra griega para mensajero, *áγγελος* y a la inteligencia artificial (IA) que posee, la cual simula mantener una conversación con una persona al proveer recomendaciones automáticas, utilizando técnicas avanzadas de Procesamiento del Lenguaje Natural (NLP, por sus siglas en inglés), como lo son los actualmente muy utilizados *transformers* (Vaswani et al., 2017), además se considera el análisis emocional para ofrecer recomendaciones de películas más precisas y personalizadas. En este sentido *AngellA* tiene la capacidad de interactuar con los usuarios, haciéndoles preguntas sobre sus estados de ánimo y gustos, y evaluará las respuestas utilizando técnicas de NLP para extraer información emocional. Este enfoque permite comparar las emociones y preferencias extraídos de las respuestas de los usuarios con las emociones y géneros previamente clasificados de las películas, lo que facilita la recomendación de películas afines a las preferencias y estados de ánimo de los usuarios.

La introducción de la arquitectura de *transformers* en el sistema de recomendación permite una mayor precisión y adaptabilidad en la captura de las preferencias y emociones de los usuarios en tiempo real, dado que estos, utilizan mecanismos de autoatención para capturar dependencias a largo plazo en secuencias de texto y han demostrado ser eficientes en una amplia gama de tareas de NLP, incluida la comprensión y generación de texto (Vaswani et al., 2017). Los transformers, han logrado el entrenamiento masivo de *corpus* de texto, los Grandes

Modelos de Lenguaje (*Large Language Models* o *LLM*) capaces de poder realizar una amplia gama de tareas de NLP como la generación de texto, por ejemplo, entrenar un chatbot a través de una instrucción (*Prompt*, en inglés) en lenguaje natural (Wei et al., 2023).

El trabajo se organiza en capítulos, comenzando por un contexto y estado de la técnica en el que se presentan los enfoques existentes en el campo de los sistemas de recomendación de películas, el análisis emocional y los *chatbots*. Se describe el diseño y la implementación del sistema propuesto, seguido de la evaluación de su rendimiento utilizando métricas y realimentación por medio de la satisfacción del usuario. Por último, se presentan las conclusiones y posibles trabajos futuros en esta área de interés.

La propuesta de este trabajo es relevante y oportuna, ya que busca abordar las limitaciones de los sistemas de recomendación de películas tradicionales y mejorar la experiencia de los usuarios, al ofrecer recomendaciones más precisas y personalizadas que consideren las emociones y preferencias. Esto tiene el potencial de aumentar el compromiso y la satisfacción de los usuarios, lo que a su vez puede conducir a un mayor disfrute del contenido y una experiencia de entretenimiento más gratificante.

La combinación de análisis emocional y *chatbots* en un sistema de recomendación de películas también tiene implicaciones más amplias para la investigación y el desarrollo de sistemas de recomendación en general, ya que podría aplicarse a otros dominios y tipos de contenido, como música, libros o videojuegos. Por tanto, este trabajo tiene como objetivo contribuir al avance del conocimiento en el campo de la IA, proporcionando un enfoque novedoso y efectivo para abordar los desafíos que enfrentan los usuarios al navegar por el vasto universo de opciones de entretenimiento disponibles en la actualidad.

1.1.Motivación

El objeto de estudio de este trabajo de fin de máster es la optimización de los sistemas de recomendación de películas, integrando las emociones de los usuarios como parte fundamental mostrando recomendaciones en tiempo real. La creciente demanda de contenido de entretenimiento y la gran cantidad de opciones disponibles (Westcott et al., 2021) hacen que sea cada vez más difícil para los usuarios encontrar películas que se ajusten a sus gustos y estados de ánimo actuales.

Las causas de este problema pueden atribuirse a las limitaciones de las técnicas de recomendación tradicionales (Roy & Dutta, 2022) como el filtrado colaborativo.

El filtrado colaborativo es una técnica comúnmente utilizada en sistemas de recomendación para generar sugerencias personalizadas para los usuarios basadas en las preferencias y comportamientos de otros usuarios con gustos similares. La idea central detrás del filtrado colaborativo es que, si dos usuarios han mostrado una similitud en sus elecciones o valoraciones pasadas de elementos, es probable que tengan intereses similares en el futuro. Por lo tanto, el sistema de recomendación utiliza esta información de similitud para predecir y recomendar elementos a uno de los usuarios que aún no ha interactuado pero que son altamente apreciados por el otro usuario con gustos afines (Fernández-Paniagua, 2018).

El filtrado colaborativo no considera factores como las emociones y las preferencias cambiantes de los usuarios. Además, la interacción entre los usuarios y los sistemas de recomendación a menudo es limitada y no permite una comunicación dinámica que ayude a refinar las recomendaciones en función de las necesidades y deseos específicos de cada usuario (Guo et al., 2019).

La relevancia del problema radica en el potencial de los sistemas de recomendación de películas para mejorar la experiencia de entretenimiento de los usuarios y su satisfacción general (Jayalakshmi et al., 2022). Al considerar las emociones y preferencias en tiempo real, los sistemas de recomendación pueden ofrecer recomendaciones más precisas y personalizadas que se ajusten a las necesidades y deseos específicos de cada usuario, lo que a su vez puede conducir a un mayor compromiso y disfrute del contenido (Jayalakshmi et al., 2022).

En este contexto, la aplicación de técnicas avanzadas de NLP, como los *transformers*, y la integración de análisis emocional en los sistemas de recomendación de películas, puede abordar algunas de estas limitaciones y mejorar la calidad de las recomendaciones. Además, al incorporar un *chatbot* en el sistema de recomendación, se puede mejorar la interacción entre los usuarios y el sistema, permitiendo una comunicación más dinámica y adaptativa que refleje las necesidades y deseos del usuario.

1.2.Planteamiento del problema

El presente trabajo se enfoca en el diseño e implementación de un sistema de recomendación de películas basado en el análisis emocional, haciendo uso de técnicas avanzadas de NLP como son los *transformers*. La solución a este desafío requiere abordar múltiples aspectos, tales como la identificación y evaluación de emociones y estados de ánimo, segmentación de películas basada en su contenido como sinopsis y reseñas de los consumidores previos, así como la integración de un *chatbot* para facilitar la interacción con el usuario.

Para solucionar este problema, se propone el siguiente enfoque:

1. Desarrollar un *chatbot* que interactúe con el usuario de manera natural y efectiva, haciendo preguntas sobre sus estados de ánimo y emociones actuales (M. McTear et al., 2016). El *chatbot* también podría proporcionar información adicional sobre las películas recomendadas y redireccionar a detalles adicionales del contenido.
2. Implementar técnicas avanzadas de NLP, como los *transformers* (Vaswani et al., 2017), para analizar y extraer información emocional y temática tanto del diálogo del usuario como de las descripciones de las películas. Esto permitiría identificar emociones y estados de ánimo en el texto, así como los temas y géneros asociados.
3. Crear un modelo de recomendación que utilice la información emocional proporcionada por el usuario para generar recomendaciones de películas personalizadas y emocionalmente coincidentes. Este modelo podría basarse en técnicas de aprendizaje automático (ML, por sus siglas en inglés), como la similitud de coseno (Javed, 2020), el aprendizaje profundo (DL, por sus siglas en inglés) (Joy, 2020) o el aprendizaje por refuerzo (Srinivasan, 2019).
4. Evaluar el rendimiento del sistema de recomendación utilizando diferentes métricas y la satisfacción del usuario (Herlocker et al., 2004; Cremonesi et al., 2011).

1.3.Estructura de la memoria

El trabajo se organiza en los siguientes apartados:

1. **Introducción:** Se presenta la motivación, el planteamiento del trabajo y la estructura del trabajo.

2. **Contexto y estado de la técnica:** se realiza una revisión de la literatura sobre sistemas de recomendación, análisis emocional, detección de emociones y *chatbots*, incluyendo trabajos relevantes en cada área.
3. **Objetivos y metodología de trabajo:** se establecen los objetivos específicos del proyecto y se describe la metodología utilizada para alcanzarlos. Se detallan los pasos y enfoques tomados para llevar a cabo cada una de las fases del trabajo.
4. **Identificación de requisitos:** se enumeran y explican detalladamente los requisitos del sistema. Se analizan las necesidades funcionales y no funcionales del sistema de recomendación basadas en análisis emocional y *chatbot*. Esto incluye aspectos como la usabilidad, la escalabilidad, la interacción con el usuario y los requisitos técnicos.
5. **Descripción de la herramienta software desarrollada:** se abordan diversos aspectos. Primero, en la creación del conjunto de datos, se detallan las variables esenciales para el análisis emocional en películas, revisando los conjuntos de datos más relevantes en repositorios públicos y sus potenciales limitaciones. En el ámbito del análisis emocional y detección de emociones, se presentan las técnicas empleadas, abarcando el análisis de emociones desde texto y el uso de modelos pre entrenados (Nandwani & Verma, 2021). El diseño e implementación del *chatbot* se expone detalladamente, destacando la elección de tecnologías, construcción de diálogos y su interacción con el usuario. Por último, se discute el proceso de integración de la base de datos de películas para el sistema de recomendación.
6. **Evaluación y mejora:** se describen las métricas utilizadas para evaluar el rendimiento del sistema de recomendación, así como la metodología para medir la satisfacción del usuario y un planteamiento de posibles formas de optimizar el sistema en función de los resultados obtenidos.
7. **Conclusiones y trabajos futuros:** se sintetizan las conclusiones alcanzadas en este trabajo. Posteriormente, se exploran direcciones prospectivas para investigaciones futuras en el ámbito de sistemas de recomendación fundamentados en análisis emocional, contemplando potenciales extensiones y refinamientos del sistema elaborado en el presente estudio.

2. Contexto y estado de la técnica

2.1. Sentimientos y emociones: definición

Los términos "sentimiento" y "emoción" han sido objeto de análisis y debate en múltiples disciplinas académicas, desde la psicología hasta la filosofía. A pesar de que con frecuencia se emplean de manera intercambiable en el lenguaje coloquial, es esencial distinguirlos para comprender a profundidad la naturaleza de la experiencia humana.

1. **Sentimientos:** son respuestas duraderas y subjetivas que se desarrollan a lo largo del tiempo en respuesta a una experiencia o evento (Barrett, 2006). Representan la valoración cognitiva de una experiencia y pueden ser positivos, negativos o neutros.

1.1. **Sentimientos positivos:** estos sentimientos se asocian con experiencias placenteras o gratificantes, como la felicidad, la alegría y la satisfacción. Pueden surgir en respuesta a eventos que cumplen con las expectativas o deseos de una persona (Ekman, 1992).

1.2. **Sentimientos negativos:** los sentimientos negativos, por otro lado, están relacionados con experiencias desagradables o perjudiciales, como la tristeza, el enojo, la ansiedad y el miedo. Estos sentimientos suelen manifestarse cuando una persona se enfrenta a situaciones que percibe como amenazantes o problemáticas (Lazarus, 1991).

1.3. **Sentimientos neutros:** los sentimientos neutros representan una falta de respuesta emocional significativa. Pueden ocurrir en situaciones donde una experiencia no genera una reacción emocional particularmente intensa, como en actividades cotidianas rutinarias (Ekman, 1992).

2. **Emociones:** son respuestas breves, intensas y específicas a estímulos internos o externos. Estas respuestas son universalmente reconocibles en la mayoría de las culturas y pueden incluir estados emocionales como el miedo, la alegría, la tristeza, la ira, la sorpresa, entre otros (Ekman, 1992). Según el diccionario de la Real Academia Española (RAE), las emociones anteriores pueden ser definidas como:

2.1. **Miedo:** derivado del latín "*metus*" que significa "temor", se manifiesta como una angustia provocada por un riesgo o daño, ya sea real o imaginario. Además, puede definirse como el recelo o aprensión que una persona experimenta ante la posibilidad de que ocurra algo contrario a sus deseos o expectativas.

- 2.2. **Alegría:** derivada de la palabra "alegre", es un sentimiento grato y vivo que a menudo se manifiesta a través de signos exteriores. Estos signos pueden traducirse en palabras, gestos o actos específicos con los cuales las personas expresan su júbilo o el estado alegre en que se encuentran.
- 2.3. **Amor:** se origina del latín "amor, -ōris" y se conceptualiza como un sentimiento intenso del ser humano que surge de su propia insuficiencia y busca la conexión y unión con otro. Se manifiesta también como una atracción natural hacia otra persona, que al aspirar a reciprocidad en la unión, nos completa y energiza para coexistir, comunicarnos y crear. Además, engloba un sentimiento de afecto y dedicación hacia alguien o algo.
- 2.4. **Tristeza:** cuyo origen etimológico proviene del latín "*tristitia*", se refiere a la cualidad de estar triste. Es un estado emocional caracterizado por sentimientos de desánimo, melancolía o pesar.
- 2.5. **Ira:** cuyo origen etimológico proviene del latín "*ira*", es un sentimiento de indignación que produce enojo en el individuo. Además, puede manifestarse como un apetito o deseo de venganza, llevando a la persona a actuar o reaccionar con intensidad ante situaciones percibidas como injustas o provocadoras.
- 2.6. **Sorpresa:** se define como la acción y efecto de sorprender, siendo una respuesta ante situaciones o estímulos inesperados. Es una emoción que se manifiesta cuando se enfrenta a algo no anticipado, ya sea un evento, una acción o incluso un objeto. Esta reacción emocional puede ser desencadenada tanto por experiencias positivas como negativas.

2.2. Neurociencia de los sentimientos y emociones

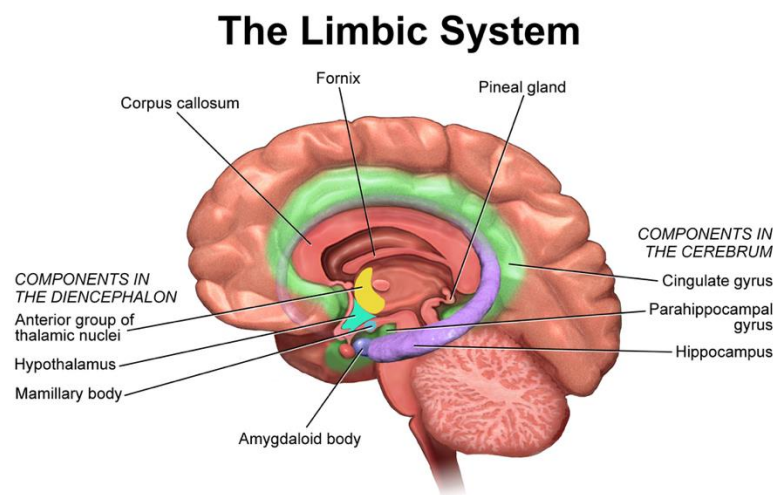
Desde una perspectiva neurocientífica, tanto los sentimientos como las emociones están vinculados a la actividad del cerebro y la liberación de neurotransmisores. Las investigaciones han identificado regiones cerebrales específicas involucradas en la experiencia emocional:

1. **Amígdala:** Desempeña un papel central en la percepción y el procesamiento de las emociones, especialmente el miedo y la amenaza (LeDoux, 1998). Esta estructura juega un papel crucial en la evaluación de la importancia emocional de los estímulos.

2. **Corteza Prefrontal:** Está involucrada en la regulación y el control de las emociones, así como en la evaluación de las consecuencias a largo plazo de las respuestas emocionales (Ochsner & Gross, 2005).
3. **Sistema Límbico:** Que incluye varias regiones cerebrales, coordina la respuesta emocional y regula las funciones fisiológicas asociadas con las emociones, como el ritmo cardíaco y la liberación de hormonas (Davis & Whalen, 2001).

La Figura 1 muestra todos los componentes del sistema límbico encargado de coordinar las respuestas emocionales.

Figura 1. Sistema límbico encargado de coordinar las respuestas emocionales.



Fuente: Blaus (2014). Licencia Creative Commons

2.2.1. Procesamiento de las señales emocionales en el cerebro

Es un proceso altamente complejo y multifacético. Diversas regiones cerebrales colaboran en la percepción, interpretación y respuesta a las señales emocionales. La percepción de las señales emocionales comienza con la detección de estímulos a través de los sentidos, como la vista y el oído. Las regiones sensoriales del cerebro, como la corteza visual y auditiva, juegan un papel fundamental en esta etapa inicial (LeDoux, 1998).

En la evaluación de la importancia emocional de los estímulos, la amígdala tiene un papel fundamental. Si un estímulo se percibe como amenazante o significativo desde el punto de vista emocional, la amígdala puede activarse, desencadenando una respuesta emocional intensa (Phelps, 2006).

Después de que las señales emocionales son procesadas por la amígdala, la corteza prefrontal, especialmente la corteza prefrontal media, se involucra en la interpretación y la asignación de significado a estas señales. Esta región está involucrada en la autoevaluación y la introspección emocional, lo que permite a las personas reflexionar sobre sus propias emociones y experiencias (Ochsner & Gross, 2005).

2.2.2. Las Vías y Circuitos Emocionales

El procesamiento de las emociones y los sentimientos se lleva a cabo a través de vías y circuitos específicos en el cerebro. Estas vías facilitan la comunicación entre diferentes regiones cerebrales involucradas en la experiencia emocional.

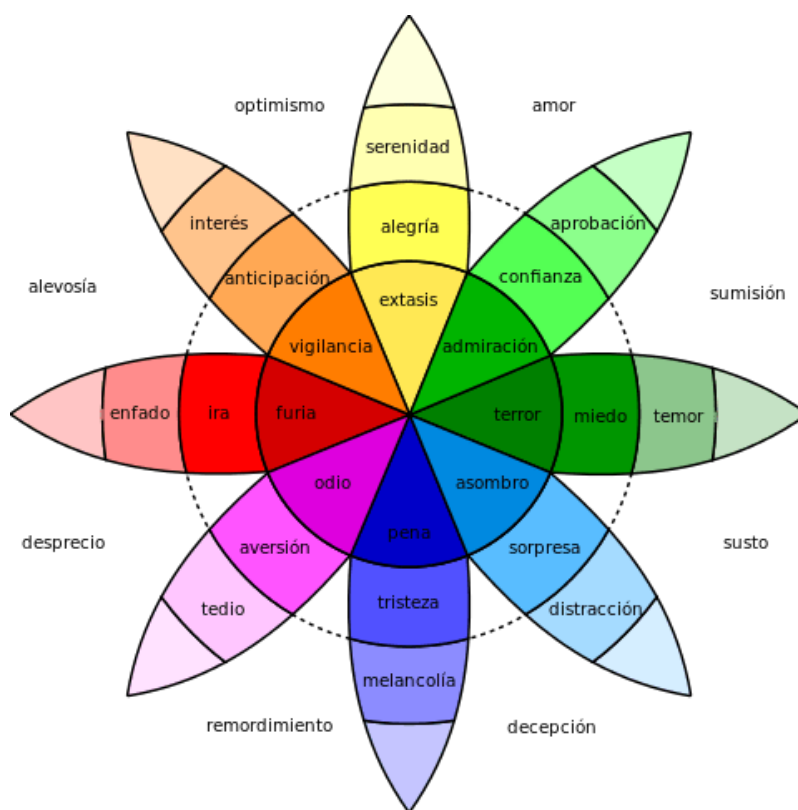
1. **Vía Tálamo-Amígdala:** El tálamo actúa como un "centro de retransmisión" que envía información sensorial a la amígdala de manera rápida y directa, lo que permite una respuesta emocional inmediata ante estímulos amenazantes (LeDoux, 1998).
2. **Vía Córtico-Amigdal:** Esta vía implica la interacción entre la corteza prefrontal y la amígdala, permitiendo una evaluación más detallada y contextualizada de las señales emocionales (Ochsner & Gross, 2005).

2.3. Avances en el estudio de emociones

En la década de 1980, la investigación de las emociones tomó un giro significativo con la formulación de la Teoría Psicoevolucionista de las Emociones (Plutchik, 1980). Esta teoría, basada en la idea que las emociones desempeñan un papel adaptativo en la evolución humana, postula que sirven como mecanismos adaptativos en respuesta a los desafíos del entorno. Uno de sus logros más notables fue la identificación de ocho emociones primarias organizadas en pares opuestos, como alegría versus tristeza, confianza versus aversión, sorpresa versus anticipación, y enojo versus temor.

Otra contribución importante fue la creación de la "rueda de las emociones", una representación gráfica que ilustra la estructura y jerarquía de las emociones (Plutchik, 1980). Esta rueda no solo destaca las emociones primarias y sus opuestos, sino que también permite visualizar las combinaciones y grados de intensidad de cada emoción, como se muestra en la Figura 2.

Figura 2. Rueda de las emociones, donde se muestra su estructura y jerarquía.



Fuente: Plutchik (1980).

Tabla 1. Diferentes emociones y sentimientos asociados a cada una de ellas.

Emociones	Ejemplos de palabras o frases indicativas	Sentimiento asociado
felicidad	feliz, contento, alegre, emocionado, eufórico, optimista.	positivo
tristeza	triste, deprimido, melancólico, desanimado, sombrío.	negativo
ira	enojado, irritado, frustrado, furioso, molesto.	negativo
miedo	asustado, temeroso, nervioso, inseguro, preocupado, ansioso.	negativo
sorpresa	sorprendido, asombrado, impactado, desconcertado, estupefacto.	neutro
asco	asqueado, repugnado, nauseabundo, desagradable, repelente.	negativo
curiosidad	curioso, intrigado, interesado, deseoso de aprender, inquisitivo.	positivo
aburrimiento	aburrido, desinteresado, apático, hastiado, cansado	neutro
relajación	relajado, tranquilo, sereno, sosegado, descansado	positivo
nostalgia	nostálgico, melancólico, anhelante, evocador, sentimental.	neutro

Fuente: elaboración propia.

En el contexto de la representación gráfica de las emociones, la Tabla 1 descompone y detalla distintas emociones, proporcionando ejemplos concretos de palabras o frases que las indican

y el sentimiento asociado a cada una, ya sea positivo, negativo o neutro. La clasificación se basa en expresiones lingüísticas comunes asociadas con cada emoción, lo que evidencia la relación entre el lenguaje y las experiencias emocionales.

En la década de 1990 y 2000, la atención sobre las emociones se centró en la investigación neurocientífica y los avances en neuroimagen, marcando un período de progreso significativo con el desarrollo de tecnologías de neuroimagen: la resonancia magnética funcional (fMRI, por sus siglas en inglés) y la tomografía por emisión de positrones (PET, por sus siglas en inglés), que desempeñaron un papel crucial en la exploración de las bases neuronales de las emociones (Phan et al., 2002).

En 2005, Paul Ekman propuso que existen emociones básicas que se manifiestan durante la realización de tareas fundamentales en la vida, y a partir de estas emociones primarias, pueden surgir otras emociones compuestas (Ekman, 2005). Además, otros modelos teóricos han contribuido al entendimiento de las emociones básicas, coincidiendo en ciertos aspectos esenciales, como la discreción de las emociones, la existencia de un conjunto fijo de expresiones corporales o neurales, y la presencia de un componente de sensación y motivación fija (Tracy & Randles, 2011). La Tabla 2 presenta algunos modelos que identifican distintas emociones básicas.

Tabla 2. Modelos para las diferentes emociones básicas propuestos por diferentes autores.

Emociones básicas soportadas teórica y empíricamente de acuerdo con cada modelo			
<i>Izard</i>	<i>Panksepp y Watt</i>	<i>Levenson</i>	<i>Ekman y Cordaro</i>
Felicidad	Juego	Disfrute	Felicidad
Tristeza	Pánico / Dolor	Tristeza	Tristeza
Miedo	Miedo	Miedo	Miedo
Ira	Furia	Ira	Ira
Asco		Asco	Asco
Interés	Buscando	Interés	
Desprecio			Desprecio
	Lujuria	Amor	
	Cuidado	Alivio	Sorpresa

Fuente: Tracy & Randles (2011).

2.4. Avances en la detección de sentimientos y emociones en texto mediante IA

La capacidad de comprender y reconocer las emociones humanas es crucial para mejorar la interacción entre las personas y las máquinas (Poria et al., 2017; Nandwani & Verma, 2021). En este contexto, el NLP ha desempeñado un papel fundamental en la extracción de información emocional y sentimental a partir de datos textuales (Pang & Lee, 2008; Liu, 2012).

Los primeros esfuerzos en la detección de emociones en texto se centraron en el análisis de sentimiento, buscando la capacidad de identificar si un fragmento de texto tiene una carga emocional positiva, negativa o neutra. Uno de los hitos tempranos fue la creación del Diccionario de Sentimiento AFINN (Nielsen, 2011), que asigna puntuaciones de sentimiento a palabras en inglés.

Otro hito significativo en esta área tuvo lugar en 2013 con la presentación del enfoque basado en Redes Neuronales Recurrentes (RNN, por sus siglas en inglés). Este enfoque contribuyó de manera fundamental a una comprensión más profunda y contextual de las emociones presentes en el texto (Socher et al., 2013).

Los modelos basados en RNN operan mediante la construcción de árboles sintácticos que representan la estructura gramatical de las frases contenidas en un texto. Cada palabra se convierte en un nodo en este árbol y se establecen conexiones entre las palabras de acuerdo con su relación gramatical en la oración. Posteriormente, se asigna un valor de sentimiento a cada palabra en función de su contexto dentro del árbol.

Lo que distingue al modelo anterior es su capacidad para capturar la información contextual y las interacciones entre palabras. En contraposición a enfoques más simples que consideran las palabras de manera aislada, los modelos basados en RNN tienen la capacidad de reconocer cómo las palabras interactúan en el contexto de una oración o un párrafo. Este enfoque posibilita la identificación de matices emocionales que antes podrían pasar inadvertidos.

Otro avance reciente fue la incorporación de modelos de atención y contexto en la detección de emociones. Estos modelos consideran no solo las palabras individuales en el texto, sino también las relaciones y conexiones entre ellas. Un ejemplo es el *Transformer*, un modelo de atención múltiple, que ha demostrado ser efectivo en el análisis de texto emocional (Vaswani et al., 2017), la evolución de este modelo a otros como BERT (Devlin et al., 2019) y GPT

(Radford et al., 2018) hicieron una revolución en la detección de sentimientos y emociones. Estos modelos, entrenados en vastas cantidades de texto, pueden transferir su conocimiento previo a tareas específicas de análisis de emociones, mejorando la precisión y la capacidad de adaptación a diferentes dominios.

2.5. Enfoques en la extracción de emociones

La tarea de extracción de emociones de texto puede considerarse como una tarea NLP de clasificación *multi label* o multi etiqueta debido a que un texto escrito puede tener diversas emociones presentes (Das, 2021).

Varios enfoques han sido propuestos para abordar el análisis de sentimientos y la detección de emociones en texto, que pueden ser clasificados en tres categorías principales: basados en léxicos, en ML y en DL (Medhat et al., 2014).

1. **Léxico:** utilizan diccionarios y listas de palabras previamente construidas y asociadas con sentimientos o emociones específicas (Hu & Liu, 2004; Strapparava & Mihalcea, 2007). Estos enfoques suelen ser simples y rápidos, pero pueden ser limitados en términos de cobertura y flexibilidad.
2. **Machine learning:** emplean técnicas de clasificación supervisada, como máquinas de vectores de soporte (SVM, por sus siglas en inglés), árboles de decisión y clasificadores Bayesianos, para predecir sentimientos y emociones a partir de características extraídas del texto (Pang et al., 2002; Kouloumpis et al., 2011). Aunque estos métodos pueden ofrecer un rendimiento mejor que los enfoques basados en léxicos, requieren datos etiquetados y pueden ser sensibles a la selección de características y la elección del algoritmo.
3. **Deep learning:** las redes neuronales convolucionales (Bharti et al., 2022) (CNN, por sus siglas en inglés) y las RNN (Majumder et al., 2018), han demostrado ser efectivas en el análisis de sentimientos y la detección de emociones debido a su capacidad para modelar representaciones de texto de alto nivel y capturar información contextual (Socher et al., 2013; Tang et al., 2015). Sin embargo, estos enfoques pueden requerir grandes cantidades de datos y recursos computacionales para entrenamiento y ajuste de modelos. (L. Zhang et al., 2018).

2.6. Desafíos en la detección de sentimientos y emociones

A medida que progresa la detección de sentimientos y emociones en texto mediante IA, también se han presentado desafíos significativos que requieren atención y resolución. Los desafíos más significativos son:

1. **Polisemia y ambigüedad:** el lenguaje humano es intrínsecamente ambiguo y polisémico, lo que significa que las palabras y frases pueden tener múltiples interpretaciones emocionales en diferentes contextos (Cruse, 1986). Esta característica fundamental del lenguaje presenta un desafío significativo en la detección de emociones a través de NLP.

Para abordar eficazmente este desafío, los investigadores han desarrollado enfoques basados en el contexto, que aprovechan la información circundante para inferir el significado emocional de una palabra o frase en un contexto específico (Reimers & Gurevych, 2019). Este enfoque contextual se ha convertido en una estrategia esencial en la detección de emociones en el lenguaje natural, permitiendo una interpretación más precisa y sensible a las variaciones de significado emocional en diferentes situaciones.

2. **Expresión sutil de emociones:** la expresión de emociones en el texto puede ser sutil y compleja, lo que añade un nivel adicional de dificultad a la detección precisa de emociones (Pang & Lee, 2008). Las personas a menudo recurren a recursos lingüísticos como metáforas, ironía y sarcasmo para comunicar sus estados emocionales de manera indirecta y matizada (Davidov et al., 2010).

La investigación en este ámbito se ha centrado en el desarrollo de modelos de NLP que puedan capturar estas expresiones sutiles de emociones en el texto (Strapparava & Mihalcea, 2007). Por ejemplo, (Davidov et al., 2010) se enfocaron en la detección de emociones a través del análisis de características lingüísticas específicas y el reconocimiento de patrones de palabras que indican la presencia de expresiones emocionales sutiles. Este enfoque ha demostrado ser efectivo para identificar emociones no explícitas en el texto, mejorando la precisión de la detección.

3. **Ética y Privacidad:** el análisis emocional de textos plantea inquietudes en el ámbito ético y de la privacidad, especialmente cuando se aborda el procesamiento de datos de carácter personal (Metzinger & Wiese, 2017). La extracción de datos delicados con

el fin de identificar las emociones de los sujetos conlleva interrogantes asociadas al consentimiento informado y la protección de la privacidad de los usuarios (Goodman & Flaxman, 2016). Investigaciones contemporáneas, como la llevada a cabo por (Metzinger & Wiese, 2017), han profundizado en las ramificaciones éticas del análisis emocional en textos, postulando directrices que abogan por una práctica ética en este dominio. Dichas directrices aspiran a asegurar que el análisis emocional se ejecute con responsabilidad, preservando los derechos y la privacidad de los sujetos implicados.

2.7. Sistemas de recomendación

2.7.1. Definición

Los sistemas de recomendación son herramientas esenciales en diversas aplicaciones, como comercio electrónico, entretenimiento y redes sociales, para ayudar a los usuarios a descubrir contenido relevante y personalizado (Ricci et al., 2011). En su esencia, es un sistema que emplea algoritmos para sugerir elementos o productos a usuarios cuando visitan un sitio web o exploran un producto en línea. Estas sugerencias pueden abarcar diversas categorías, desde productos que visualizar y adquirir hasta películas que ver, anuncios para visualizar e información para leer. Su funcionalidad es esencial en plataformas de publicidad en línea y en el comercio electrónico, donde ayudan a los usuarios a descubrir productos que puedan ser de su interés (Mbaabu, 2021).

La base fundamental de estos sistemas de recomendación radica en su capacidad para analizar y entender las preferencias de los usuarios, su actividad previa, y los tipos de retroalimentación que proporcionan. Para lograr esto, se basan en cuatro tipos principales de información: actividad previa del usuario, preferencias, retroalimentación implícita y retroalimentación explícita (Melville & Sindhvani, 2011).

La retroalimentación implícita se obtiene a través de las acciones de los usuarios, como las compras, clics y búsquedas que realizan. Por otro lado, la retroalimentación explícita se basa en las reacciones de los usuarios, como sus "me gusta" o "no me gusta" hacia productos específicos. Estas dos formas de retroalimentación son esenciales para que el sistema de recomendación comprenda las preferencias y necesidades de los usuarios (Melville & Sindhvani, 2011).

2.7.2. Taxonomía de los sistemas de recomendación

Antes de implementar un sistema de recomendación, es esencial determinar qué tipo de sistema se desea desarrollar. Una forma de comenzar es observar sistemas similares como inspiración. Una taxonomía propuesta en *Practical Recommender Systems* (Falk, 2019), describe un sistema en función de varias dimensiones como dominio, propósito, contexto, nivel de personalización, opiniones, privacidad y confiabilidad, interfaces y algoritmos. Es importante considerar cada una de estas dimensiones al desarrollar un sistema de recomendación.

1. **Dominio:** se refiere al tipo de contenido que se recomienda. Puede abarcar una amplia variedad de contenidos, películas, libros, vehículos, entre otros.

La importancia del dominio radica en que proporciona indicaciones sobre cómo se utilizarán las recomendaciones. Además, señala el grado de precisión necesario en las recomendaciones. Asimismo, determina si es posible recomendar el mismo contenido en más de una ocasión.

2. **Propósito:** el propósito de un sistema de recomendación aborda tanto las necesidades del usuario final como las del proveedor. Para el usuario, el sistema de recomendaciones busca ofrecer contenido relevante que desee consumir en un momento específico. Sin un sistema de ordenamiento o filtrado adecuado, sería desafiante encontrar contenido en una extensa biblioteca. Por otro lado, para el proveedor, el objetivo es retener a los usuarios ofreciéndoles contenido de interés constantemente.

Un sistema de recomendación no se limita únicamente a la venta o consumo de contenido. Puede tener objetivos educativos, informativos o de ayuda al usuario. Sin embargo, en muchos casos, el propósito principal es incrementar las ventas o la retención de usuarios.

3. **Contexto:** se refiere al entorno en el que el consumidor recibe una recomendación, puede estar determinado por factores como el dispositivo utilizado por el usuario, su ubicación actual, el momento del día y las actividades que está realizando. Otros factores contextuales pueden incluir el clima o incluso el estado de ánimo del usuario.
4. **Nivel de personalización:** el grado de personalización en las recomendaciones puede variar desde el uso de estadísticas básicas hasta el análisis de datos individuales de los usuarios.

- 4.1. **No personalizado:** se basan en listas de los artículos más populares, suponiendo que el usuario actual podría tener preferencias similares a la mayoría. Este tipo de recomendaciones también puede incluir listas ordenadas por fecha, donde se muestran los elementos más recientes primero. Todos los usuarios que interactúan con el sistema de recomendación reciben la misma lista.
- 4.2. **Semi personalizado:** el siguiente nivel de recomendaciones segmenta a los usuarios en grupos, proporcionando recomendaciones semi personalizadas. Los usuarios pueden ser segmentados de diversas maneras, como por edad, nacionalidad o patrones distintivos de comportamiento o preferencias.
- Estos sistemas de recomendación no poseen información personal sobre el usuario, solo lo identifican como miembro de un grupo o segmento. Las recomendaciones se basan en la pertenencia a estos segmentos y, por lo tanto, los usuarios que pertenecen al mismo segmento recibirán las mismas sugerencias.
- 4.3. **Personalizado:** las recomendaciones personalizadas se basan en datos específicos del usuario, reflejando cómo ha interactuado previamente con el sistema. Estas recomendaciones se generan específicamente para cada usuario individual. Aunque la mayoría de los sistemas de recomendación utilizan segmentación y popularidad al crear recomendaciones personalizadas, hay sitios que ofrecen recomendaciones totalmente personalizadas basadas en la interacción y preferencias del usuario. Además, algunos sitios combinan varios tipos de recomendaciones, desde las más genéricas hasta las más personalizadas, basadas en la interacción del usuario con determinados productos o servicios.
5. **Privacidad y confiabilidad:** se debe considerar cómo un sistema protege la privacidad de los usuarios y cómo se utiliza la información recopilada.
- Los sistemas de recomendación deben mantener la confianza del usuario. Si los usuarios comienzan a sentirse manipulados perderán la confianza en las recomendaciones y buscarán alternativas en otro lugar. Además, las recomendaciones que tienen el poder de influir en las decisiones se convierten en objetivos para actores maliciosos con intenciones no genuinas.
- La confiabilidad se refiere a cuánto confía el consumidor en las recomendaciones, en lugar de verlas como intentos comerciales o de manipulación. Un sistema es confiable si el usuario toma en serio las sugerencias proporcionadas.

6. **Interfaz:** describe el tipo de entrada y salida que produce. A continuación, se examina cada una de estas categorías.
 - 6.1. **Entrada:** históricamente, ciertos sistemas permitían a los usuarios introducir sus gustos y aversiones a través de calificaciones y preferencias sobre géneros y temas. Esta información puede utilizarse como entrada para un sistema de recomendación. La entrada puede ser explícita, donde el usuario introduce manualmente información sobre sus preferencias, o implícita, donde el sistema deduce los gustos basándose en la interacción con el usuario.
 - 6.2. **Salida:** las formas de salida pueden ser predicciones, recomendaciones o filtrado. Algunos sistemas presentan estimaciones, proporcionan sugerencias personalizadas y muestran elementos populares, generalmente en forma de lista de los más destacados.

Algunos sistemas explican las recomendaciones que ofrecen. Aquellos que ofrecen una explicación son llamados recomendadores de caja blanca, mientras que aquellos que no lo hacen son denominados recomendadores de caja negra. Cuanto más necesita un sistema explicar, más sencillo es el algoritmo y viceversa.
7. **Algoritmo:** desempeñan un papel crucial en los sistemas de recomendación. Se pueden clasificar en varias categorías según el tipo de datos que emplean para generar recomendaciones. Estas categorías incluyen filtrado colaborativo, filtrado basado en contenido y recomendadores híbridos.

2.7.3. Categorías de sistemas de recomendación

Estos sistemas pueden ser clasificados en dos categorías principales: filtrado colaborativo y filtrado basado en contenido (Adomavicius & Tuzhilin, 2005).

1. **Filtrado colaborativo:** se basa en la idea de que los usuarios que han mostrado preferencias similares en el pasado también mostrarán preferencias similares en el futuro (Resnick et al., 1994). Estos sistemas pueden ser de dos tipos: basados en memoria y basados en modelos (Koren et al., 2009). Los sistemas basados en memoria utilizan medidas de similitud, como la correlación de Pearson o la similitud del coseno.

Correlación de Pearson: la correlación de Pearson es una tasa o ratio entre la covarianza de dos variables. En general, indica un grado de correlación lineal en un conjunto de datos (x, y) , donde 1 implica una correlación perfectamente lineal y 0 implica que no existe una correlación lineal significativa (Pearson, 1895).

Matemáticamente, la correlación de Pearson entre dos variables (x, y) puede calcularse así:

$$r = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^n (x_i - \bar{x})^2} \sqrt{\sum_{i=1}^n (y_i - \bar{y})^2}} \quad (1)$$

Donde:

n , es el total de pares de datos (x, y) .

\bar{x} , es la media de todos los valores de la variable x .

\bar{y} , es la media de todos los valores de la variable y .

Similitud de coseno: la similitud del coseno es una medida de la similitud entre dos vectores de un mismo espacio dimensional. Matemáticamente, dos vectores son similares si el coseno del ángulo entre ellos es cercano a cero, un valor de 1 implica que los vectores son distantes.

Por definición, la similitud del coseno se calcula como:

$$\text{sim}(x, y) = \cos(\vec{x}, \vec{y}) = \frac{\vec{x} \cdot \vec{y}}{\|\vec{x}\| \|\vec{y}\|} = \frac{\sum_{i=1}^n x_i y_i}{\sqrt{\sum_{i=1}^n x_i^2} \sqrt{\sum_{i=1}^n y_i^2}} \quad (2)$$

Donde:

n , es la dimensión de los vectores \vec{x}, \vec{y} .

$\|\vec{x}\|$ y $\|\vec{y}\|$, es la norma o magnitud de los vectores \vec{x}, \vec{y} .

La técnica de similitud de coseno puede utilizarse para medir la similitud en textos luego de convertir los documentos a una representación vectorial, como, por ejemplo,

las técnicas de *bag of words (BoW)* (Gunawan et al., 2018). También, pueden emplearse para identificar usuarios o elementos similares y generar recomendaciones en función de las calificaciones o interacciones anteriores de estos usuarios o elementos (Sarwar et al., 2001).

Por otro lado, los sistemas basados en modelos que emplean técnicas de ML, como la factorización matricial y los modelos de mezcla de Gaussianas, para aprender patrones y relaciones subyacentes en los datos de interacción del usuario y predecir las preferencias futuras, también hacen uso de las técnicas de similitud descritas anteriormente (Salakhutdinov et al., 2007; Rendle et al., 2012).

2. **Filtrado de contenido:** se centra en las características de los elementos y utiliza la similitud de contenido entre elementos para generar recomendaciones (Pazzani & Billsus, 2007). Estos sistemas suelen emplear técnicas de NLP y recuperación de información para extraer y modelar características relevantes, como palabras clave, temas y géneros, a partir de descripciones de elementos (Mooney & Roy, 1999; Lops et al., 2011). Aunque el filtrado basado en contenido puede ser eficaz para abordar problemas como el inicio en frío, puede estar limitado en términos de diversidad y ser propenso a la sobre especialización (Adomavicius & Tuzhilin, 2005).

Una comparativa de los métodos, mostrando sus características, ventajas y desventajas para sistemas de recomendación se puede observar en la Tabla 3.

2.7.1. Modelado de un sistema de recomendación basado en NLP

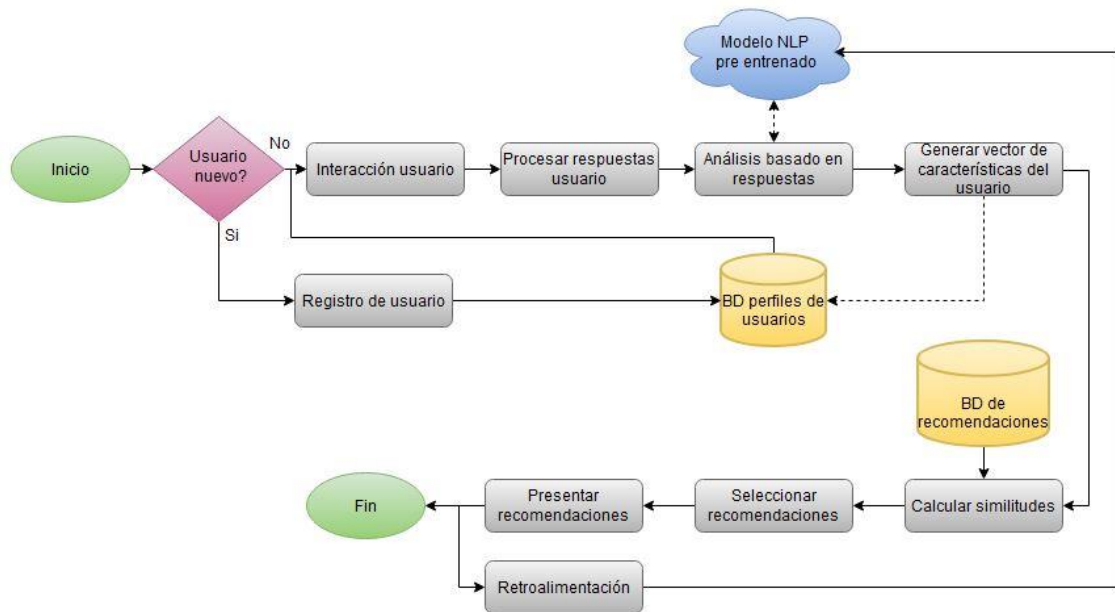
Figura 3 ilustra el diagrama de flujo de un sistema de recomendación fundamentado en NLP. Al ingresar al sistema, se verifica la existencia previa del usuario en una base de datos. Si el usuario es identificado como nuevo, se procede a crear un registro correspondiente. Una vez completado este paso, el usuario interactúa con el sistema de recomendación. A partir de estas interacciones, se lleva a cabo un análisis basado en las respuestas obtenidas. El modelo NLP, previamente entrenado, formula un vector que encapsula características propias de cada usuario, que luego es comparado con los perfiles asociados a las recomendaciones mediante un cálculo de similitud. A partir de esta comparación, se elaboran las recomendaciones específicas para el usuario. Como etapa final, el modelo NLP recibe retroalimentación para optimizar y perfeccionar futuras recomendaciones.

Tabla 3. *Comparativo de diferentes enfoques de sistemas de recomendación.*

Método	Características	Ventajas	Limitaciones
Basado en contenido	Utiliza características de los elementos (películas) para recomendar elementos similares.	<ul style="list-style-type: none"> - Personalización - No requiere datos de otros usuarios - Aprendizaje incremental 	<ul style="list-style-type: none"> - Dificultad para extraer características - Limitado a elementos similares
Filtrado colaborativo	<ul style="list-style-type: none"> - Filtrado colaborativo basado en usuarios - Filtrado colaborativo basado en elementos 	<ul style="list-style-type: none"> - Buen rendimiento en sistemas grandes - Capacidad de descubrir intereses implícitos - Robustez frente a cambios en las preferencias de usuario 	<ul style="list-style-type: none"> - Problema de inicio en frío - Problema de dispersión
Basado en conocimiento	Utiliza conocimiento del dominio y reglas heurísticas para hacer recomendaciones en función de necesidades.	<ul style="list-style-type: none"> - No necesita datos históricos - Útil en dominios especializados 	<ul style="list-style-type: none"> - Necesita conocimiento previo del dominio - Mantenimiento y actualización del conocimiento
Híbrido	Combina dos o más enfoques de recomendación	<ul style="list-style-type: none"> - Mayor precisión y diversidad - Soluciona problemas de inicio en frío y dispersión 	<ul style="list-style-type: none"> - Mayor complejidad - Mayor tiempo de desarrollo y mantenimiento

Fuente: elaboración propia.

Figura 3. Diagrama de flujo para sistema de recomendación usando NLP.



Fuente: elaboración propia.

2.8. Chatbots en sistemas de recomendación

La incorporación de *chatbots* en sistemas de recomendación ha ganado popularidad en los últimos años debido a su capacidad para interactuar con los usuarios de manera natural y en tiempo real (Jannach et al., 2022). Los *chatbots* pueden mejorar la experiencia de usuario y facilitar la exploración y el descubrimiento de contenido, al tiempo que proporcionan información adicional y responden a preguntas relacionadas con el contenido recomendado (McTear et al., 2016).

1. **Chatbots basados en reglas:** los *chatbots*, desde su concepción inicial, han evolucionado notablemente en su complejidad y aplicaciones. Uno de los primeros enfoques en la construcción de *chatbots* fue el modelo basado en reglas. Estos *chatbots* funcionan mediante un conjunto predefinido de reglas que dictan cómo el *chatbot* debe responder ante entradas específicas de los usuarios (Weizenbaum, 1966). Dicho de otro modo, para cada entrada específica del usuario, existe una respuesta predefinida por parte del *chatbot*. El *chatbot* ELIZA, creado por Joseph Weizenbaum en el *Massachusetts Institute of Technology* (MIT), es uno de los ejemplos más emblemáticos de este tipo de sistema, simulando una terapeuta empática y proporcionando respuestas basadas en scripts preprogramados.

Una de las ventajas más destacadas de los *chatbots* basados en reglas es su consistencia y fiabilidad. Al operar bajo un conjunto predefinido de instrucciones, estos *chatbots* ofrecen respuestas coherentes y predecibles cada vez, lo que garantiza una experiencia de usuario uniforme (Shawar & Atwell, 2007). Esto los hace especialmente útiles en aplicaciones donde la consistencia es primordial, como atención al cliente o gestión de pedidos.

Sin embargo, no están exentos de limitaciones. Una desventaja significativa es su falta de flexibilidad (McTear, 1998). A diferencia de los *chatbots* impulsados por la IA que pueden aprender de nuevas entradas y adaptarse con el tiempo, los *chatbots* basados en reglas están confinados a su programación original. Si un usuario plantea una consulta que no está prevista en las reglas del *chatbot*, este último puede no ser capaz de proporcionar una respuesta adecuada, lo que puede frustrar al usuario.

Además, a medida que las necesidades de interacción se vuelven más complejas, el mantenimiento y la actualización de un *chatbot* basado en reglas pueden resultar laboriosos. La adición de nuevas reglas o la modificación de las existentes puede requerir una revisión extensa del sistema y potencialmente introducir errores o inconsistencias (Shawar & Atwell, 2007).

2. ***Chatbots* basados en técnicas de DL:** el uso de técnicas avanzadas de NLP, como los *transformers* en *chatbots* ha sido un área de investigación activa (Vaswani et al., 2017). Los *transformers* han demostrado ser efectivos en una amplia gama de tareas de NLP, incluido el análisis de sentimientos, la respuesta a preguntas y la generación de texto, gracias a su capacidad para capturar información contextual a largo plazo y modelar representaciones semánticas de alto nivel (Devlin et al., 2018; Radford et al., 2019). En el contexto de sistemas de recomendación, los *chatbots* basados en *transformers* pueden ser utilizados para extraer información emocional y temática de las interacciones del usuario y utilizar esta información para generar recomendaciones personalizadas y emocionalmente conscientes (Wolf et al., 2020). Además, estos *chatbots* pueden ser fácilmente adaptados y ajustados a dominios específicos, como el entretenimiento y las películas, mediante la transferencia de aprendizaje y la adaptación de dominio (Howard & Ruder, 2018).

Recientemente, los LLM con billones de parámetros pre entrenados en *corpus* gigantescos de texto (GPT, Llama, PALM, etc.) (Wei et al., 2023) han demostrado

habilidades para generar respuestas coherentes y contextuales, mejorando significativamente la calidad y la naturalidad de las interacciones con los usuarios.

Los LLM generalmente son comparados entre sí por su número de parámetros, donde usualmente más es mejor. El número de parámetros también define la complejidad del modelo, y tiene una relación directa con la cantidad de datos de los que puede aprender y generar texto. Sin embargo, el consumo de recursos es mayor. (Maeda & Bolaños, 2023).

Tabla 4. *Crecimiento de parámetros por año para los LLM.*

LLM	Parámetros	Año
BERT	340M	2018
GPT-2	1.5M	2019
Meena	2.6M	2020
GPT-3	175B	2020
LaMDA	137B	2022
BLOOM	176B	2022

Fuente: (Maeda & Bolaños, 2023).

El entrenamiento de *chatbots* por medio de LLM también ha representado una simplificación en la cantidad de código necesario para su implementación, debido a que los LLMs pueden entrenarse en base a instrucciones (*prompts*). Esta reciente técnica, aun en estudio es conocida generalmente como *prompt engineering* y permite entrenar al *chatbot* sobre el tipo de respuestas que debe proporcionar y el comportamiento que debe mostrar en una conversación, produciendo diálogos llenos de mucho contexto. (Arora et al., 2022).

No obstante, persisten desafíos en cuanto a la detección de sesgos y alucinaciones del modelo dependiendo del *corpus* de texto con el que ha sido entrenado. Comunidades médicas han expresado su preocupación debido a que los *chatbots* entrenados con LLMs pueden diagnosticar incorrectamente, lo que conlleva a un peligro en la salud de las personas. (Gilbert et al., 2023). Todo esto sugiere la necesidad de investigaciones adicionales para perfeccionar su desempeño y garantizar su implementación ética y efectiva en diversas aplicaciones comerciales y sociales.

2.8.1. Arquitectura de un sistema de chatbot

Independientemente del modelo de *chatbot* que se elija, la arquitectura de un sistema que integre *chatbot* y recomendaciones junto a análisis emocional se presenta en la Figura 4. En primer lugar, la interacción se inicia cuando un usuario se comunica con el sistema a través de una interfaz de usuario o una aplicación de *chat*. Esta interfaz sirve como punto de entrada y facilita la comunicación bidireccional entre el usuario y el sistema de chatbot, garantizando que las consultas del usuario se procesen y que las respuestas se transmitan de manera efectiva al usuario.

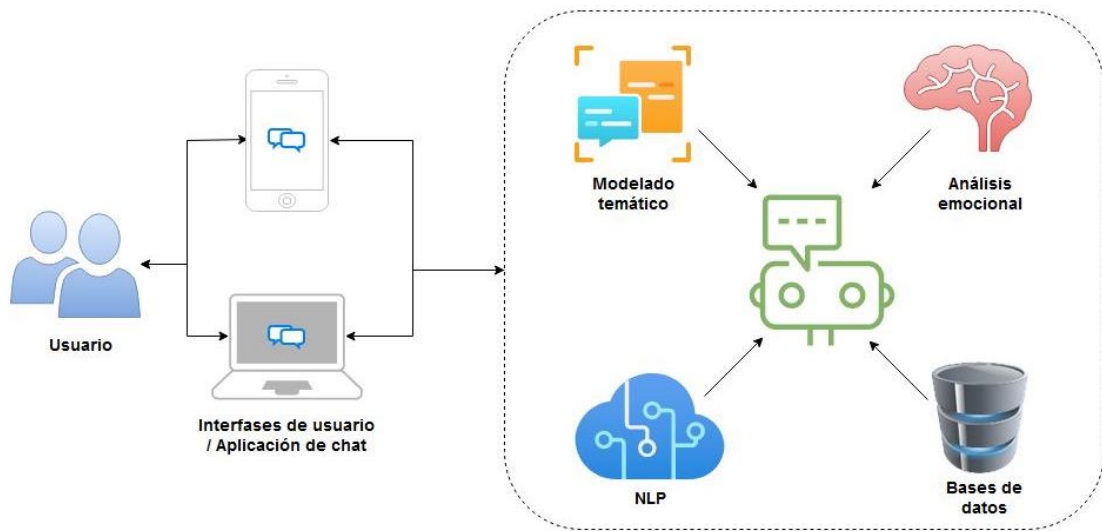
La arquitectura de un sistema de chatbot o de una aplicación en general, consta de dos componentes fundamentales: *backend* y *frontend*, ampliamente conocidos por su terminología en inglés. El backend, es responsable de gestionar y procesar los datos, así como de realizar las operaciones lógicas y de negocio detrás de la aplicación. Por otro lado, el frontend es la capa visible para el usuario y se encarga de presentar la información de manera comprensible y de interactuar con el usuario a través de la interfaz gráfica (Chapaval, 2017).

En el núcleo de la arquitectura, varios componentes trabajan de manera integrada para procesar y responder a las consultas del usuario. El componente de NLP se encarga de interpretar el lenguaje humano y convertirlo en un formato comprensible para el sistema. La base de datos almacena información relevante y datos históricos que el *chatbot* puede necesitar para formular respuestas. El análisis emocional evalúa el tono y el contexto emocional de las consultas del usuario para proporcionar respuestas adecuadas a su estado emocional. Finalmente, el modelado temático ayuda al sistema a identificar y comprender el tema principal de la consulta del usuario, garantizando respuestas más precisas y contextualizadas.

2.9. Desafíos y oportunidades en la investigación

A pesar del progreso en el análisis de sentimientos, sistemas de recomendación y *chatbots*, aún existen desafíos y oportunidades en la investigación que deben abordarse para desarrollar un sistema de recomendación de películas emocionalmente “*consciente*” y personalizado (Lin et al., 2023).

Figura 4. Esquema básico de la arquitectura de un chatbot.



Fuente: elaboración propia.

Algunos de estos desafíos incluyen:

1. La representación eficiente y precisa de las emociones y estados de ánimo en el texto, así como la identificación de sus relaciones con los temas y géneros de las películas (Calvo & D'Mello, 2010; Binali et al., 2010). Esto podría implicar la exploración de nuevos enfoques y modelos para el análisis de sentimientos y la detección de emociones, así como la adaptación y ajuste de modelos existentes al dominio de las películas (Ibrahim et al., 2019).
2. La integración de la información emocional y temática en modelos de recomendación que puedan tener en cuenta tanto las preferencias explícitas como implícitas de los usuarios y generar recomendaciones que sean relevantes, diversas y emocionalmente adecuadas (Z. Zhang et al., 2019). Esto podría requerir la investigación de nuevas técnicas y enfoques para la combinación de filtrado colaborativo y basado en contenido, así como la adaptación de modelos de DL y aprendizaje por refuerzo al problema de recomendación emocionalmente consciente (Yuyan et al., 2019).
3. El diseño y la implementación de *chatbots* que puedan interactuar de manera efectiva y natural con los usuarios, adaptándose a sus estados de ánimo, emociones y preferencias en tiempo real (M. McTear et al., 2016). Esto implica la investigación de nuevos enfoques y técnicas para la generación de diálogos y la adaptación de dominio,

así como la evaluación y mejora continua del rendimiento y la satisfacción del usuario (Caldarini et al., 2022).

3. Objetivos y metodología de trabajo

3.1. Objetivo general

Desarrollar un *chatbot* para recomendación de películas personalizadas que utilice enfoques avanzados de NLP, que logre una comprensión y análisis de las emociones y preferencias del usuario, proporcionando recomendaciones altamente relevantes y precisas.

3.2. Objetivos específicos

1. Investigar y analizar las técnicas y algoritmos de NLP más adecuados para el análisis de sentimientos y la comprensión del contexto en la interacción con usuarios.
2. Revisar y comparar diferentes enfoques y arquitecturas de sistemas de recomendación de películas, incluidos los métodos basados en contenido, filtrado colaborativo y enfoques híbridos.
3. Desarrollar e implementar un modelo de clasificación de emociones para películas, basado en la información textual y las opiniones de los usuarios.
4. Diseñar e implementar un *chatbot* que utilice técnicas de NLP y el modelo de clasificación de emociones para generar recomendaciones de películas basadas en el estado de ánimo y los gustos del usuario.
5. Evaluar la efectividad y precisión del *chatbot* para recomendar películas mediante pruebas con usuarios y análisis de sus respuestas y satisfacción en función de las recomendaciones recibidas.

3.3. Metodología del trabajo

Para alcanzar los objetivos propuestos en este trabajo, se seguirá una metodología estructurada en varias etapas, detallando los pasos a seguir en cada una de ellas, así como los instrumentos y recursos necesarios para su ejecución. A continuación, se describe la metodología del trabajo:

1. **Revisión bibliográfica:** en esta etapa se llevará a cabo una revisión exhaustiva de la literatura científica y académica relacionada con NLP, el análisis de sentimientos, los

sistemas de recomendación de películas y los *chatbots*. Se consultarán diversas fuentes, como bases de datos académicas, publicaciones en conferencias y revistas especializadas, y libros de texto relevantes en el área. Los resultados de esta revisión bibliográfica se utilizarán para fundamentar y respaldar las decisiones y enfoques tomados en las siguientes etapas.

2. **Selección de técnicas y algoritmos:** a partir de la revisión bibliográfica, se seleccionarán las técnicas y algoritmos de NLP más adecuados para el análisis de sentimientos y la comprensión del contexto en la interacción con usuarios. Se prestará especial atención a las técnicas basadas en *transformers* y otras arquitecturas de DL que han demostrado un buen rendimiento en tareas similares.
3. **Creación del modelo de clasificación de emociones:** en esta etapa, se desarrollará e implementará un modelo de clasificación de emociones para películas, utilizando la información textual y las opiniones de los usuarios. Se recopilarán y procesarán conjuntos de datos relevantes, se entrenará y ajustará el modelo, y se evaluará su rendimiento utilizando métricas apropiadas.
4. **Diseño e implementación del *chatbot*:** con el modelo de clasificación de emociones desarrollado en la etapa anterior, se diseñará e implementará un *chatbot* que utilice técnicas de NLP y el modelo de clasificación de emociones para generar recomendaciones de películas basadas en el estado de ánimo del usuario. Se desarrollarán módulos de interacción con el usuario, análisis de las respuestas y generación de recomendaciones utilizando el modelo de clasificación de emociones.
5. **Evaluación del *chatbot* para recomendar películas:** en esta etapa, se llevará a cabo la evaluación de la efectividad y precisión del *chatbot* para recomendar películas mediante pruebas con usuarios y análisis de sus respuestas y satisfacción en función de las recomendaciones recibidas. Se recopilarán y analizarán datos cualitativos y cuantitativos, y se calcularán métricas de rendimiento para evaluar la calidad de las recomendaciones y la experiencia del usuario.
6. **Análisis de resultados y conclusiones:** por último, se analizarán los resultados obtenidos en las etapas anteriores y se elaborarán conclusiones con respecto al

cumplimiento de los objetivos del trabajo. Se identificarán las limitaciones del estudio, así como posibles mejoras y futuras líneas de investigación relacionadas con el tema.

La metodología propuesta se basa en un enfoque iterativo y flexible (SCRUM). SCRUM se fundamenta en iteraciones denominadas "*sprints*", con duración fija y secuencias de actividades bien definidas, que promueven la entrega incremental de funcionalidades. SCRUM, como marco adaptable y autoorganizado, potencia la eficiencia, transparencia y adaptabilidad de sistemas de IA, lo que permite realizar ajustes y refinamientos en cada etapa según sea necesario (Ameta et al., 2023).

4. Identificación de requisitos

Como se mencionó anteriormente, el problema a tratar consiste en diseñar un sistema de recomendaciones de películas que parta del análisis emocional de los usuarios para generar recomendaciones más precisas. Dada la naturaleza interactiva entre el sistema de recomendación y los usuarios, se realizará un *chatbot* que genere recomendaciones de películas basado en todo el contexto de la conversación y la base de datos de reseñas de las películas.

A continuación, se enumeran y explican una serie de requisitos identificados para la realización del sistema de *software* a desarrollar.

4.1. Conjunto de datos de las películas

Un aspecto relevante para el desarrollo de la investigación es encontrar o construir un conjunto de datos que contenga variables que permitan generar un modelo clasificatorio de emociones adaptado a películas. Opcionalmente, puesto que la mayoría de las investigaciones y avances de NLP es en inglés (Wali et al., 2020), podría considerarse la viabilidad de construir un conjunto de datos en español para aportar al avance de las investigaciones de NLP en este idioma.

Las variables del conjunto de datos pueden abarcar las siguientes (pero no necesariamente ser las únicas):

1. **Título de la película:** Esta variable permitirá diferenciar las películas de todo el conjunto de datos, además que posibilitará realizar consultas con una base de datos

para identificar y extraer información adicional sobre la película y luego ser entregada al usuario en la interacción en tiempo real.

2. **Sinopsis de la película:** Esta variable contendrá una descripción lo suficientemente concisa de la película, dado que se utilizará para la identificación de similitudes semánticas entre todas ellas y el contexto de la conversación del usuario, para que posteriormente se pueda confirmar si se considera acertada la recomendación o solicitar una nueva. Además, usando técnicas de clusterización se facilitará la creación de agrupaciones significativas que trasciendan las categorías de género convencionales, con el propósito de brindar una comprensión más profunda y diversificada de las relaciones entre las producciones cinematográficas.

El análisis de similitudes semánticas en la sinopsis se fundamenta en el concepto de que las películas pueden compartir elementos narrativos, temáticos o emocionales que trascienden las etiquetas de género tradicionales. A través de la identificación de estas similitudes, se pretende trazar conexiones conceptuales que, de otro modo, podrían pasar desapercibidas.

3. **Reseñas (*reviews*, en inglés):** El propósito de esta variable es encontrar una media de las emociones principales que puedan detectarse en los comentarios de las personas que han visto anteriormente la película. De esta manera puede compararse la emoción de un usuario del *chatbot* con la emoción detectada en las reseñas de las películas. Las reseñas de usuarios constituyen una manifestación genuina de las respuestas individuales y subjetivas hacia una película. Esta dimensión subjetiva captura tanto las impresiones emocionales positivas como negativas que cada película evoca en los espectadores, aportando una gama completa de matices emocionales. Posteriormente, mediante el empleo de técnicas de NLP y análisis de emociones, será posible cuantificar y categorizar estas respuestas emocionales en términos de su polaridad y su relación con el contenido cinematográfico.

El análisis de reseñas puede desvelar un amplio espectro de emociones que se vinculan con distintos aspectos de las películas. Es posible identificar patrones recurrentes en las respuestas emocionales, lo que podría llevar a la agrupación de películas con similitudes emocionales en clústeres específicos.

4. **Género:** La relevancia de esta variable se deriva de su capacidad para proporcionar una amplia gama de recomendaciones potenciales al usuario, garantizando así una

oferta diversificada acorde con distintas preferencias cinematográficas. Los distintos géneros cinematográficos tienen el potencial de evocar diferentes tipos de emociones en la audiencia.

5. **Clasificación (*rating*):** Esta variable es esencial para adecuar las recomendaciones cinematográficas al rango de edad del usuario, garantizando así propuestas apropiadas y pertinentes según su perfil demográfico.
6. **Año de lanzamiento:** Esta variable permite contextualizar la película dentro de un marco temporal específico para su identificación única en el conjunto de datos.

4.2. Modelos de Deep Learning para la detección de emociones

Es una práctica común utilizar técnicas de *Transfer Learning* (TL) en diversas tareas de NLP (Ruder et al., 2019) debido a la dificultad de entrenar un modelo desde cero; el alto costo de recursos computacionales sumado a los prolongados tiempos de ejecución y la dificultad de conseguir grandes cantidades de datos para entrenamiento, requieren que se aborde el tema desde otras perspectivas (Teslenko, 2023).

TL permite adaptar un modelo entrenado previamente en un conjunto de datos y utilizar los pesos (*weights*) del modelo para ocuparlos en otra tarea relacionada a la tarea original, pero con distintas entradas (*inputs*) al modelo (Tunstall et al., 2022). Las estrategias de TL se han vuelto más atractivas a medida que fuentes de datos a gran escala se vuelven cada vez más accesibles. La utilización de conjunto de datos disponibles que están de alguna manera relacionados, pero no idénticos, convierte a este enfoque de aprendizaje en un método viable y capaz de mejorar las métricas de rendimiento de modelos tradicionales de ML (Hosna et al., 2022).

Existen distintos repositorios abiertos (*open source*) de internet que contienen modelos pre entrenados para poder utilizarse en diversos tipos de tareas de DL en general (Fadel, 2022), de las cuales es importante destacar *Hugging Face* (Jain, 2022) al ser una comunidad que permite construir, entrenar y poner en marcha diversas aplicaciones de IA utilizado principalmente por investigadores, científicos de datos e ingenieros de ML para compartir ideas, modelos *open source* y proporcionar ayuda. De esta manera *Hugging Face* contribuye al avance y la democratización de la IA permitiendo además una fácil interacción con todos

los modelos alojados de la comunidad vía código a través de sus librerías en *Python* (Mahood, 2022).

Una exploración en el sitio de *Hugging Face* muestra más de 1500 modelos clasificatorios de emociones pre entrenados en distintos *corpus* e idiomas, que pueden servir como punto de partida para el entrenamiento del modelo a usar en este trabajo.

4.3. Traducción automática para inputs del usuario

Debido a la amplia investigación de diversas tareas de NLP en inglés y a la relativamente inferior cantidad de investigaciones en español, será necesario disponer de un modelo de traducción automática que pueda procesar el texto de entrada de los usuarios y generar traducciones de alta calidad de español a inglés manteniendo el mayor contexto posible de las oraciones originales para posteriormente ser procesadas en las diversas funciones del sistema.

La librería *Deep Translator* para *Python* se presenta como una herramienta robusta y versátil en el ámbito de la traducción automática (Krukowski, 2022). Destaca por su capacidad de interactuar con diversas API de traducción, abarcando nombres tan reconocidos como *Google*, *Microsoft* y *Yandex*, además de otros como *Libre*. Una característica que la distingue es la posibilidad de realizar traducciones directamente desde un archivo de texto. Por otro lado, la facilidad de uso también se ve reflejada en su integración con la terminal, permitiendo operaciones de traducción directamente desde la misma. Asimismo, posee la capacidad de llevar a cabo traducciones en lote, optimizando tiempos en proyectos de gran escala. Sin embargo, es pertinente señalar que la actividad y el soporte para su mantenimiento depende de la comunidad activa actualmente. Este panorama refleja una herramienta con un alto potencial, pero con algunos aspectos que requerir atención o consideración detallada al momento de implementarla en este proyecto.

4.4. Frameworks para la creación de chatbots

Actualmente, existen diversos marcos de trabajo (de aquí en adelante, *framework*, en inglés, al ser el termino mayormente utilizado en la comunidad de programadores) (Gupta, 2022) que facilitan la creación y puesta en producción de *chatbots* de diversas categorías, ya sean de índole general, capaces de sostener una conversación muy variada hasta aquellos

especializados en un solo tema. Estos *frameworks* proporcionan una diversidad de herramientas cuyo objetivo es reducir el tiempo de creación de un sistema de *chatbot*.

Las soluciones que se implementan pueden estar enfocadas desde la creación de un *chatbot* como un programa como servicio (SaaS, Software as a Service, por sus siglas en inglés) donde el proveedor de la infraestructura es el encargado del desarrollo de las acciones que realizará el *chatbot* (Radziwill & Benton, 2017) mientras que el desarrollador (o cliente del *framework*) establece las características que debe cumplir el *chatbot*, además se encarga de las pruebas y verificar la efectividad del sistema creado. Algunos de los *frameworks* principales que ofrecen SaaS son *DialogFlow* de *Google* (Muhammad et al., 2020), en *Microsoft Power Virtual Agents* (Greyling, 2019) y *Watson* de *IBM* (Kumar et al., 2022) entre otros.

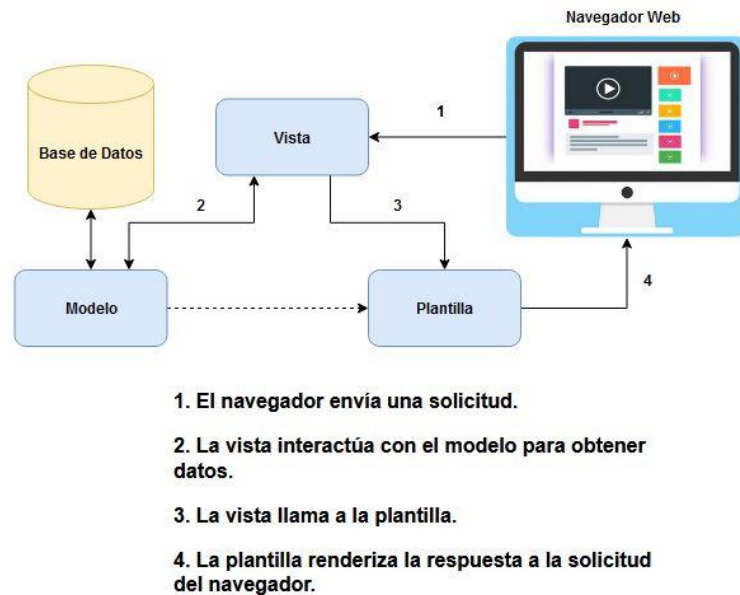
Soluciones complejas permiten el desarrollo de un producto de *software* más personalizado en la que el *framework* solo se encarga de establecer un marco de buenas prácticas a seguir en la implementación del sistema final, como *Langchain* (Topsakal & Akinci, 2023) o *Rasa NLU* (Jiao, 2020).

Otra de estas soluciones es *Django* (William et al., 2023). Este es un *framework* de desarrollo *web* basado en *Python*, que es una plataforma robusta para la configuración de la infraestructura de un sistema de *chatbot*. Su arquitectura, fundamentada en el patrón MTV (Model Template View, por sus siglas en inglés) (ver Figura 5), permite una relación entre la estructura de datos, la lógica de negocio y la presentación al usuario, otorgando una implementación modular que resulta esencial para sistemas escalables. Esta estructuración facilita tanto el mantenimiento como las futuras ampliaciones o adaptaciones que el *chatbot* pueda requerir. Además, *Django* cuenta con un conjunto integrado de herramientas que simplifican tareas esenciales como la administración de bases de datos, la autenticación de usuarios y la garantía de seguridad, componentes vitales para un *chat* interactivo que aspire a resguardar la integridad de los datos y la privacidad del usuario.

Adicionalmente, uno de los grandes activos de *Django* es su extensa documentación y la solidez de su comunidad. Estos recursos no solo proporcionan soluciones a problemáticas comunes, sino que impulsan la creación de extensiones (o plugins, en inglés) que pueden ser incorporados al sistema de *chatbot* para expandir sus funcionalidades. Y no menos importante es la versatilidad de *Django* en su capacidad para integrarse con otras librerías y *frameworks* de *Python*. Esto es especialmente relevante cuando se considera la necesidad de incorporar

herramientas de NLP o motores de IA en el *chatbot*, haciendo de *Django* una opción adecuada para diseñar sistemas de *chat* avanzados y altamente funcionales.

Figura 5. Arquitectura MTV de Django.



Fuente: elaboración propia.

4.5. Características del flujo de conversación

Independientemente del *framework* y librería con la que se implemente el desarrollo final del *chatbot*, será necesario que cumpla una serie de requisitos:

1. **Debe cumplir con las características establecidas en la norma ISO 9241:** según Radziwill & Benton (2017) dada la naturaleza de interactividad con el usuario, el *chatbot* deberá cumplir con las normas referentes a la usabilidad de sistemas de *software*. Lo cual puede resumirse en *eficiencia*: en su desempeño a la hora de proporcionar un sistema a prueba de entradas no deseadas por parte del usuario, *efectividad*: en la facilidad de uso y en la flexibilidad de adaptarse a las necesidades propias de cada usuario y *satisfacción*: en el sentido que sea capaz de generar una conversación agradable con el usuario.
2. **Limitación adecuada de la experticia del sistema:** la rapidez y eficiencia del sistema implica limitarlo a un área específica, en el caso particular de recomendación de películas debe limitarse cualquier otra interacción no relacionada y dirigir al usuario a dicho fin (Meyer, 2023).

3. **Idioma de entrada:** se requiere desarrollar un *chatbot* con la capacidad de mantener diálogos en español, lo que implica la necesidad de abordar aspectos como la traducción automática. En este contexto, la utilización de librerías especializadas cobra relevancia al posibilitar la adecuada interpretación de las consultas realizadas por los usuarios, lo que a su vez permite ofrecer respuestas precisas y pertinentes. Estas librerías juegan un rol fundamental al facilitar la interacción con el modelo subyacente, posibilitando una comprensión más profunda y matizada de las emociones presentes en las interacciones.
4. **Seguridad y privacidad:** sí el *chatbot* maneja información sensible de los usuarios es necesario implementar medidas de seguridad para proteger los datos y garantizar la confidencialidad de las conversaciones.

La naturaleza centrada en los datos de los *chatbots* plantea importantes preocupaciones de privacidad, con el riesgo de capturar, retener y potencialmente exponer información sensible del usuario de manera inadvertida. Esta preocupación ha sido planteada en casi todos los documentos de investigación actualizados con respecto al uso de *chatbots* (Sebastian, 2023).
5. **Rendimiento:** la optimización del rendimiento constituye una etapa de gran relevancia a considerar en el desarrollo del *chatbot*. Un *chatbot* veloz y eficiente es fundamental para proporcionar una experiencia de usuario altamente satisfactoria, especialmente en el contexto de interacciones en tiempo real (Gnewuch et al., 2018). Sin embargo, la velocidad debe ser la adecuada para que simule efectivamente la forma en la que los humanos desarrollan una conversación. En este sentido, la búsqueda de equilibrio entre la complejidad funcional y la eficiencia de ejecución se presenta como una consideración importante, con el propósito de proporcionar una experiencia de usuario fluida y satisfactoria en todo momento.

4.6. Interfaces de interacción para chatbots

La construcción de una Interfaz Gráfica de Usuario (GUI, por sus siglas en inglés) que sea agradable visualmente al usuario también es otro de los aspectos a considerar en la realización del sistema (Jansen, 1998).

La interfaz deberá ser intuitiva y de fácil uso, adecuada para usuarios con diferentes niveles de habilidades técnicas. Es esencial que los elementos gráficos y controles se dispongan

siguiendo un diseño coherente y estandarizado para lograr una experiencia de usuario fluida; la creación de flujos de conversación es otro aspecto clave de la GUI ya que debe facilitarse al usuario el acceso al historial de la conversación.

A continuación, y de acuerdo con Ramming et al. (2022) se agrupan las características de diseño o *Design Features* (DF), ideales que deberá cumplir la GUI en base a su función en el proceso de conversación:

1. Design features que implican el proceso de entrada de texto al sistema:

- 1.1. Posibilidad de respuestas rápidas para una navegación más rápida y menos esfuerzo de escritura por parte de los usuarios usando botones y otros elementos de interactividad.
- 1.2. Proporcionar contexto conversacional a través de elementos separados de la GUI (por ejemplo, botones, campos de texto) que permitan manipular el contexto. Esto implica que la GUI debe facilitar al usuario en todo momento el historial de conversación.
- 1.3. Proporcionar respuestas rápidas dinámicas que tengan en cuenta el contexto de la conversación y muestren al usuario posibles pasos siguientes. El usuario no siempre necesita ingresar texto ya que pueden presentársele alternativas a elegir dependiendo del momento en el que se encuentra en la conversación. Esta característica depende de la naturaleza del *chatbot* y no necesariamente se debe tener en cuenta.
- 1.4. Utilizar elementos multimedia como imágenes, GIF y videos en el *widget* de *chat* para transmitir información de manera visual. Será importante dotar de expresividad al *chatbot* incluyendo elementos como emojis y dando la posibilidad que el usuario consulte las películas recomendadas en base de datos de películas externas.
- 1.5. Reducir los esfuerzos de escritura mediante la implementación de una función de predicción de texto (autocompletado) que proporcione sugerencias de palabras e intenciones. En dispositivos móviles, esta opción se presenta por defecto en el teclado.

2. Design features que aportan al flujo conversacional:

- 2.1. Proporcionar información sobre el contexto de la conversación en las respuestas del *chatbot* para establecer un entendimiento común entre el usuario y el *chatbot*. Lo

cual implica restringir las respuestas del usuario a las necesarias para proporcionar la recomendación.

2.2. En caso de malentendidos, proporcionar al usuario el mensaje original para que el usuario pueda interpretar adecuadamente el mensaje. En este punto también es posible entregar una reformulación de la pregunta al usuario.

2.3. Al iniciar la conversación, presentar el *chatbot* como una entidad artificial y proporcionar al usuario las capacidades del *chatbot* para ayudar a gestionar las expectativas del usuario.

4.7. Computación en la nube

La computación en la nube ha surgido como una solución tecnológica fundamental en las últimas décadas, ofreciendo una plataforma escalable, flexible y económicamente eficiente para alojar y gestionar aplicaciones y datos (Jamsa, 2012). Se define como la entrega de servicios informáticos a través de la internet, donde los recursos, como el almacenamiento y la potencia de procesamiento, se proveen de manera dinámica según las necesidades del usuario, sin que este tenga que preocuparse por la infraestructura subyacente. En el contexto de sistemas de recomendación, particularmente aquellos que requieren análisis emocional, la computación en la nube puede proveer la capacidad de procesamiento y almacenamiento necesaria para manejar y analizar grandes volúmenes de datos, garantizando respuestas en tiempo real y optimizando la experiencia del usuario final.

Tres de los principales proveedores de servicios en la nube son: *Microsoft Azure*, *Amazon Web Services* (AWS, por sus siglas en inglés) y *Google Cloud Platform* (GCP, por sus siglas en inglés). Cada uno de estos pueden ser aprovechados en la construcción de un sistema de recomendación.

GCP ofrece ventajas en términos de precios altamente competitivo y sostenible, especialmente para proyectos que escalan rápidamente. Su red global garantiza latencias bajas y una rápida transferencia de datos, lo que es crucial para aplicaciones en tiempo real (Marinescu, 2013).

Desde una perspectiva de seguridad, GCP emplea una capa de seguridad de *hardware* personalizado, el *Titan Security Chip*, que verifica la integridad del *hardware* y *software*, protegiendo contra amenazas de suplantación y asegurando la cadena de confianza desde el

inicio. Adicionalmente, GCP utiliza técnicas avanzadas de encriptación tanto en tránsito como en reposo. Su modelo de seguridad "defensa en profundidad" asegura los datos en múltiples niveles, restringiendo el acceso no autorizado (Google, 2020).

5. Descripción de la herramienta software desarrollada

5.1. Creación del conjunto de datos de películas

La adecuada preparación de un conjunto de datos constituye un paso esencial en el proceso de construcción de un modelo clasificatorio destinado a identificar las emociones percibidas en películas por usuarios previos. Si bien en la internet se encuentran disponibles numerosos repositorios públicos de los cuales es posible recolectar diversos conjuntos de datos de películas, resulta importante ejercer un criterio selectivo al momento de elegir aquellos que alberguen las variables pertinentes para los propósitos de la investigación.

La selección de un conjunto de datos funcional para la investigación se procedió a buscar de distintas fuentes y seleccionar la que mejor reflejará los objetivos planteados.

5.1.1. Exploración de los diferentes conjuntos de datos de películas disponibles en la Internet

Los siguientes conjuntos de datos fueron los seleccionados para la investigación con el propósito de analizar las variables contenidas en cada uno de ellos, sus nombres se presentan en el idioma original de publicación (inglés) con el propósito de facilitar futuras referencias e investigaciones.

1. **MovieLens**: el conjunto de datos *MovieLens* (GroupLens, 2023) es reconocido en el campo de la recomendación de películas. Proporciona una amplia variedad de películas con diferentes atributos y dimensiones. Incluye tanto las valoraciones de los usuarios como metadatos adicionales. Estas calificaciones permiten desarrollar modelos de recomendaciones personalizadas, complementados con información sobre género, director, elenco y año de publicación.

Un submodelo es *MovieLens 25M* que contiene una amplia gama de información y consta de 25 millones de calificaciones, relacionadas con 62.000 películas y 162.000 usuarios. Este conjunto de datos fue liberado en diciembre de 2019 y se encuentra disponible para su acceso en la internet. Su tamaño es de 250 MB.

Otro conjunto de datos es *MovieLens Tag Genome 2021* proporciona aproximadamente 10.5 millones de datos de relevancia de películas y sus respectivas etiquetas (1.084 etiquetas aplicadas a 9.734 películas). Este conjunto fue lanzado en diciembre de 2021 y se encuentra disponible para su descarga en la internet. Su tamaño es de 1.8 GB.

Los conjuntos de datos mencionados previamente albergan atributos para la construcción y evolución del sistema de recomendación. A pesar de esto, es relevante señalar que carecen de información detallada en forma de reseñas de usuarios y sinopsis de las películas. Debido a esta carencia de datos críticos para el enfoque del sistema, se determinó que los conjuntos de datos pertenecientes a *MovieLens* no cumplen con los requisitos necesarios y, en consecuencia, han sido excluidos para el desarrollo de este proyecto.

2. ***Internet Movie Database (IMDb) 50K***: el conjunto de datos de IMDb (Maas et al., 2011), una autoridad de referencia en el universo cinematográfico ofrece un recurso importante para la realización de tareas de NLP y análisis de texto, al contener un total de 50.000 reseñas cinematográficas.

Este conjunto de datos está especialmente diseñado para la clasificación binaria de sentimientos y sobrepasa en tamaño a conjuntos de datos de referencia previos. El repositorio incluye un conjunto de 25.000 reseñas altamente polarizadas destinadas al entrenamiento, junto con otras 25.000 destinadas a pruebas.

Generalmente, este conjunto de datos se utiliza con el objetivo de predecir reseñas positivas y negativas empleando algoritmos de clasificación o DL. Sin embargo, para los propósitos de este trabajo ha sido descartado debido a que no incluye muchas de las variables descritas anteriormente y necesarias para el desarrollo del sistema de recomendación.

3. ***The Movie Database (TMDB) API***: la interfaz de programación de aplicaciones (API, por sus siglas en inglés) de TMDB es también una fuente de información detallada sobre películas, su interfaz programática provee acceso a metadatos de películas e incluyen aspectos cruciales como título, género, resumen, año de lanzamiento, duración y más (Pardo, 2019). La API se distingue por la amplitud de los datos, incluyendo elementos como reseñas, imágenes y *trailers*, que agregan un componente visual y descriptivo a la información proporcionada. Esta capacidad de enriquecer los datos con elementos

visuales y narrativos provee una experiencia más inmersiva y completa para los usuarios en general.

TMDB 5000 *Kaggle*: el conjunto de datos TMDB 5000 de *Kaggle* ofrece una fuente de datos que incluye información sobre la trama, el reparto, el equipo, el presupuesto y los ingresos de miles de películas. Presenta información detallada de los actores y actrices, presentándolos en el orden en que aparecen en los créditos de la película. Además, contiene campos no comunes, como "*homepage*", "*id*", "*original_title*", "*overview*", "*popularity*", "*production_companies*", "*production_countries*", "*release_date*", "*spoken_languages*", "*status*" y "*tagline*", ampliando la información disponible para clasificación.

Esta variedad de campos que potencialmente podrían resultar valiosos para el desarrollo del proyecto. Sin embargo, es relevante señalar que la ausencia de reseñas y sinopsis cinematográficas dentro de este conjunto de datos limita su idoneidad para abordar el problema planteado. La carencia de esta información podría afectar la capacidad del sistema de recomendación para proporcionar un análisis completo en el contexto de la investigación.

4. **Conjunto de datos *Netflix Prize*:** concebido como el motor detrás de un concurso destinado a mejorar las capacidades de recomendación en la plataforma de transmisión de video (*streaming*, en inglés) *Netflix*, este conjunto de datos (Bennett & Lanning, 2007) encapsula un registro de las interacciones entre usuarios y contenido. Incluye calificaciones proporcionadas por los usuarios.

El conjunto de datos contiene más de 100 millones de calificaciones de más de 480 mil usuarios anónimos de *Netflix* seleccionados aleatoriamente, en relación con más de 17 mil títulos cinematográficos. Los datos fueron recopilados entre octubre de 1998 y diciembre de 2005, y representan la distribución de todas las calificaciones recibidas durante este periodo. Estas calificaciones se encuentran en una escala de 1 a 5. Para preservar la privacidad de los usuarios, cada usuario ha sido reemplazado por un identificador asignado al azar. Además de las calificaciones, se proporciona la fecha de cada calificación, así como el título y el año de lanzamiento de cada película.

Este conjunto de datos dispone de una serie de atributos propicios para la implementación de técnicas de filtrado colaborativo, similar a las utilizadas por *Netflix*. La presencia de estos campos relacionados con el filtrado colaborativo agrega un enfoque diferente a la

versatilidad del conjunto de datos, pero se ajusta más a escenarios distintos a los propuestos en esta investigación.

5. **Conjunto de datos IMDB – IEEE Movie reviews:** este conjunto de datos (Pal et al., 2020) incluye casi 1 millón de reseñas provenientes de 1.150 películas distintas registradas en IMDb. Estas películas abarcan 17 géneros diversos presentes en IMDb, tales como *acción*, *aventura*, *animación*, *biografía*, *comedia*, *crimen*, *drama*, *fantasía*, *historia*, *terror*, *música*, *misterio*, *romance*, *ciencia ficción*, *deporte*, *thriller* y *guerra*. Adicionalmente, también proporciona metadatos relacionados con las películas, como la fecha de lanzamiento, duración, calificación IMDb, clasificación de la película por edad (PG-13, R, etc.), número de evaluadores en IMDb y la cantidad de reseñas por película.

La gran cantidad de reseñas por película presente en este conjunto de datos hace que sea uno de los más adecuados para el proyecto. A pesar de no contar con algunos campos mencionados al principio, será de importancia la personalización de este conjunto de datos agregando nuevos campos y eliminando aquellos no necesarios.

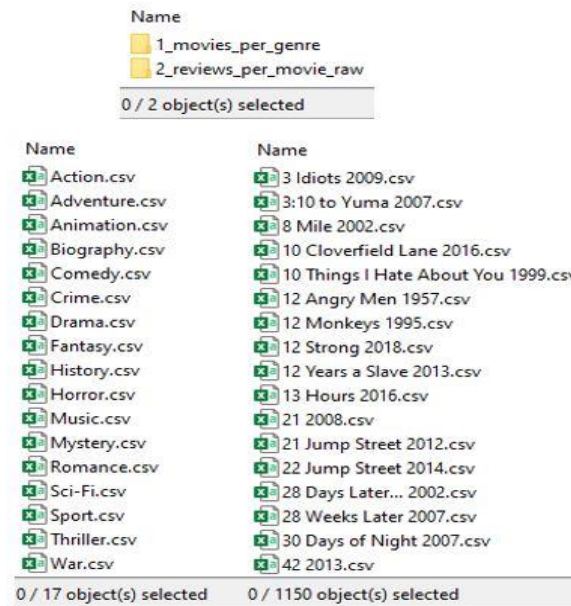
5.1.2. Preparación del conjunto de datos personalizado

Después de la selección del conjunto de datos *IMDB – IEEE Movie reviews* y su descarga desde la página de IEEE¹, se realizó un proceso de identificación de todos los archivos que lo componen. Este sistema de ficheros contiene dos carpetas, donde la primera contiene información correspondiente a un género cinematográfico distinto, como se puede apreciar en la Figura 6.

La segunda carpeta alberga un total de 1.150 archivos. Cada uno de estos archivos se nombra en función de la película y su año de lanzamiento, y contiene las respectivas reseñas de cada una. Todos los archivos del conjunto de datos están en formato CSV.

¹Enlace conjunto de datos IEEE Movie reviews: <https://ieee-dataport.org/open-access/imdb-movie-reviews-dataset>

Figura 6. Estructura del sistema de archivos del conjunto de datos TMDb.



Fuente: elaboración propia.

Para llevar a cabo el procesamiento de los datos, se procedió a la carga de los archivos CSV que contenían la información relacionada con el género de cada película en un *notebook* de *Jupyter* usando *Python*. Se creó una tabla de datos (*dataframe*, en inglés) individual para cada archivo, utilizando la librería *Pandas*. Este enfoque permitió estructurar y organizar la información contenida en cada uno de los archivos de manera efectiva.

En el código mostrado en la Figura 7, se observa la creación de un *dataframe* para cada género específico, como "*crime*", "*action*", "*adventure*", "*animation*", y así sucesivamente. Cada *dataframe* se generó utilizando la función *pd.read_csv()* de *Pandas*, que carga los datos desde el archivo CSV correspondiente y los almacena en la estructura de *dataframe*.

Luego se ejecuta un proceso de fusión (*merge*) entre todos los *dataframes*, con el objetivo de consolidar y combinar los datos contenidos en un único archivo CSV. Este proceso conlleva a la creación del archivo "peliculas.csv", que reúne información relevante sobre diversas películas, abarcando múltiples géneros cinematográficos.

Figura 7. Creación de *dataframes* por género de películas.

```
df1 = pd.read_csv('1_movies_per_genre/crime.csv')
df5 = pd.read_csv('1_movies_per_genre/action.csv')
df6 = pd.read_csv('1_movies_per_genre/Adventure.csv')
df7 = pd.read_csv('1_movies_per_genre/Animation.csv')
df8 = pd.read_csv('1_movies_per_genre/Biography.csv')
df9 = pd.read_csv('1_movies_per_genre/Comedy.csv')
df10 = pd.read_csv('1_movies_per_genre/Drama.csv')
df11 = pd.read_csv('1_movies_per_genre/Fantasy.csv')
df12 = pd.read_csv('1_movies_per_genre/History.csv')
df13 = pd.read_csv('1_movies_per_genre/Horror.csv')
df14 = pd.read_csv('1_movies_per_genre/Music.csv')
df15 = pd.read_csv('1_movies_per_genre/Mystery.csv')
df16 = pd.read_csv('1_movies_per_genre/Romance.csv')
df17 = pd.read_csv('1_movies_per_genre/Sci-Fi.csv')
df18 = pd.read_csv('1_movies_per_genre/Sport.csv')
df19 = pd.read_csv('1_movies_per_genre/Thriller.csv')
df20 = pd.read_csv('1_movies_per_genre/War.csv')
```

Fuente: elaboración propia.

En la Figura 7, se observa que se realiza una fusión en cadena, comenzando por el *dataframe* "df1" y fusionándolo con los *dataframes* posteriores, como "df5", "df6" y así sucesivamente hasta "df20". Cada fusión es efectuada utilizando la función *pd.merge()* de *Pandas*, creando un nuevo *dataframe* llamado *df_merged*, con la modalidad "outer" que conserva todos los datos y combina las filas correspondientes según las columnas seleccionadas.

A continuación, se depuraron los datos duplicados en *df_merged* empleando la función *drop_duplicates()*. Las columnas "name" y "year" fueron las seleccionadas para llevar a cabo dicho proceso. Finalmente, el *dataframe* depurado y sin duplicados, denominado "*df_merged_sin_duplicados*", se exportó como un archivo CSV con el nombre "peliculas.csv". Este archivo contiene las variables: 'name', 'year', 'movie Rated', 'run_length', 'genres', 'release_date', 'rating', 'num_raters', 'num_reviews', 'review_url'.

Una vez obtenido el *dataframe* consolidado de películas, se procedió a realizar una etapa adicional al proceso mediante la implementación de un código que posibilita la extracción y recopilación de las reseñas correspondientes a cada una de las 1.150 películas. Las reseñas de cada película se recopilan y se incorporan a un nuevo *dataframe* llamado *df_reviews*.

De cada película, se obtuvieron detalles específicos como el nombre, el año de estreno y la calificación. Mediante la combinación del título de la película y su año de lanzamiento, se conformó la variable "nombre", la cual facilita la identificación del archivo CSV con las reseñas correspondientes.

Figura 8. Creación de todos los dataframes basado en las columnas detalladas con el parámetro “on”.

```

df_merged = pd.merge(
    pd.merge(
        pd.merge(
            pd.merge(
                pd.merge(
                    pd.merge(
                        pd.merge(df1, df5, on=['name', 'year', 'movie_rated',
                                'run_length', 'genres', 'release_date', 'rating', 'num_raters',
                                'num_reviews', 'review_url'], how='outer'), df6, on=['name', 'year', 'movie_rated',
                                'run_length', 'genres', 'release_date', 'rating', 'num_raters',
                                'num_reviews', 'review_url'], how='outer'), df7, on=['name', 'year', 'movie_rated',
                                'run_length', 'genres', 'release_date', 'rating', 'num_raters',
                                'num_reviews', 'review_url'], how='outer'), df8, on=['name', 'year', 'movie_rated',
                                'run_length', 'genres', 'release_date', 'rating', 'num_raters',
                                'num_reviews', 'review_url'], how='outer'), df9, on=['name', 'year', 'movie_rated',
                                'run_length', 'genres', 'release_date', 'rating', 'num_raters',
                                'num_reviews', 'review_url'], how='outer'), df10, on=['name', 'year', 'movie_rated',
                                'run_length', 'genres', 'release_date', 'rating', 'num_raters',
                                'num_reviews', 'review_url'], how='outer'), df11, on=['name', 'year', 'movie_rated',
                                'run_length', 'genres', 'release_date', 'rating', 'num_raters',
                                'num_reviews', 'review_url'], how='outer'), df12, on=['name', 'year', 'movie_rated',
                                'run_length', 'genres', 'release_date', 'rating', 'num_raters',
                                'num_reviews', 'review_url'], how='outer'), df13, on=['name', 'year', 'movie_rated',
                                'run_length', 'genres', 'release_date', 'rating', 'num_raters',
                                'num_reviews', 'review_url'], how='outer'), df14, on=['name', 'year', 'movie_rated',
                                'run_length', 'genres', 'release_date', 'rating', 'num_raters',
                                'num_reviews', 'review_url'], how='outer'), df15, on=['name', 'year', 'movie_rated',
                                'run_length', 'genres', 'release_date', 'rating', 'num_raters',
                                'num_reviews', 'review_url'], how='outer'), df16, on=['name', 'year', 'movie_rated',
                                'run_length', 'genres', 'release_date', 'rating', 'num_raters',
                                'num_reviews', 'review_url'], how='outer'), df17, on=['name', 'year', 'movie_rated',
                                'run_length', 'genres', 'release_date', 'rating', 'num_raters',
                                'num_reviews', 'review_url'], how='outer'), df18, on=['name', 'year', 'movie_rated',
                                'run_length', 'genres', 'release_date', 'rating', 'num_raters',
                                'num_reviews', 'review_url'], how='outer'), df19, on=['name', 'year', 'movie_rated',
                                'run_length', 'genres', 'release_date', 'rating', 'num_raters',
                                'num_reviews', 'review_url'], how='outer'), df20, on=['name', 'year', 'movie_rated',
                                'run_length', 'genres', 'release_date', 'rating', 'num_raters',
                                'num_reviews', 'review_url'], how='outer')
    ), df_merged_sin_duplicados, on=['name', 'year'])
df_merged_sin_duplicados = df_merged.drop_duplicates(subset=['name', 'year'])
df_merged_sin_duplicados.to_csv('peliculas.csv', index=False, sep='A')

```

Fuente: elaboración propia.

Por último, `"df3"` se concatenó con `"df_reviews"`, que acumula todas las reseñas de las películas. La función `"reset_index()"` se utilizó para reiniciar los índices del *dataframe* resultante. La Figura 8 detalla el código descrito previamente.

El archivo final se guardó bajo el nombre “*reviews.csv*” y contiene un total de 932.464 registros de reseñas sobre las películas. El conjunto de datos final se muestra en la Figura 9.

Figura 9. Combinación de las características principales de las películas con sus reseñas.

```
df_reviews = pd.DataFrame()
contador = 1
for row in df_merged_sin_duplicados.iterrows():
    name = str(row[1]['name'])
    year = ''
    if ':' in name:
        name = name.replace(':', '_')
    if '?' in name:
        name = name.replace('?', '_')
    nombre = name + ' ' + str(row[1]['year'])
    df3 = pd.read_csv('2_reviews_per_movie_raw/' + nombre + '.csv')
    df3['name'] = str(row[1]['name'])
    df3['year'] = str(row[1]['year'])
    df3['movie_rated'] = str(row[1]['movie_rated'])
    df3['genres'] = str(row[1]['genres'])
    df_reviews = pd.concat([df_reviews, df3]).reset_index(drop=True)
    print(contador, nombre)
    contador += 1
```

```
1 The Dark Knight 2008
2 Pulp Fiction 1994
3 The Godfather 1972
4 Se7en 1995
5 The Silence of the Lambs 1991
6 The Departed 2006
7 The Wolf of Wall Street 2013
```

Fuente: elaboración propia.

Figura 10. Conjunto de datos final con las características principales de las películas y sus reseñas.

	username	rating	helpful	total	date	title	review	name	year	movie_rated	genres
0	MR_Heraclius	10	85	92	12 February 2020	The Dark Knight	Confidently directed, dark, brooding, and pack...	The Dark Knight	2008	PG-13	Action; Crime; Drama;
1	Smells_Like_Cheese	10	760	941	20 July 2008	The Batman of our dreams! So much more than a...	I got to see The Dark Knight on Wednesday nigh...	The Dark Knight	2008	PG-13	Action; Crime; Drama;
2	straightblaster	10	3746	4780	9 July 2008	Film surpasses the hype!	We've been subjected to enormous amounts of hy...	The Dark Knight	2008	PG-13	Action; Crime; Drama;
3	Aaron_Kyle	9	259	325	12 October 2018	Superhero Genre - Alive and kicking!	First I'd controversially like to point out th...	The Dark Knight	2008	PG-13	Action; Crime; Drama;
4	filmquestint	10	532	687	20 July 2008	Heath Ledger's Dark and Brilliant Swan Song!	I couldn't believe "The Dark knight" could liv...	The Dark Knight	2008	PG-13	Action; Crime; Drama;
...
932459	ernelson2002	7	0	3	9 November 2004	Bowie's relationship with Travis, and flashba...	his coughing was a little over the top, too. i...	The Alamo	2004	PG-13	Drama; History; War;
932460	peteurb-1	6	0	4	8 October 2004	The movie was interrupted with the loss of Rus...	I understand that Russel Crowe was originally ...	The Alamo	2004	PG-13	Drama; History; War;
932461	TheMovieMark	Null	0	3	14 June 2004	Not what it could've been!	"The Alamo" is a good, not great, movie. It di...	The Alamo	2004	PG-13	Drama; History; War;
932462	floridawar	Null	0	1	14 April 2004	History Not Artistry!	I enjoyed this movie a great deal, although it...	The Alamo	2004	PG-13	Drama; History; War;
932463	whamilton1	Null	0	1	6 April 2004	If you like watching Dennis Quaid think...	then this is the movie for you. Just because t...	The Alamo	2004	PG-13	Drama; History; War;

932464 rows x 11 columns

Fuente: elaboración propia.

5.2. Creación del modelo para el análisis de emociones

Como se mencionó en la Sección 2, la clasificación de textos es una de las tareas más comunes en el NLP. Y se refiere a la tarea de analizar y categorizar automáticamente piezas de texto en diferentes grupos o clases predefinidas. En otras palabras, es un proceso mediante el cual un sistema computacional examina un fragmento de texto y decide a qué categoría específica pertenece, basándose en patrones lingüísticos, palabras clave y características del contenido.

En el contexto de este estudio en particular, surge la necesidad de desarrollar un modelo de clasificación emocional capaz de identificar y categorizar las emociones presentes en las respuestas de los usuarios durante su interacción con el *chatbot*. De igual manera, se busca aplicar este modelo a las reseñas de películas contenidas en el conjunto de datos seleccionado (Pal et al., 2020). La importancia de esta tarea radica en su utilidad para generar recomendaciones de películas personalizadas y pertinentes para cada usuario.

La habilidad de comprender y categorizar las emociones expresadas por los usuarios durante sus conversaciones con el *chatbot* es esencial para mejorar la experiencia general de la interacción con este. Este modelo de clasificación de emociones puede identificar si un usuario se siente alegre, con rabia, temeroso o cualesquiera otras emociones descritas anteriormente, lo que permite al *chatbot* adaptar sus recomendaciones para satisfacer las necesidades emocionales y contextuales del usuario.

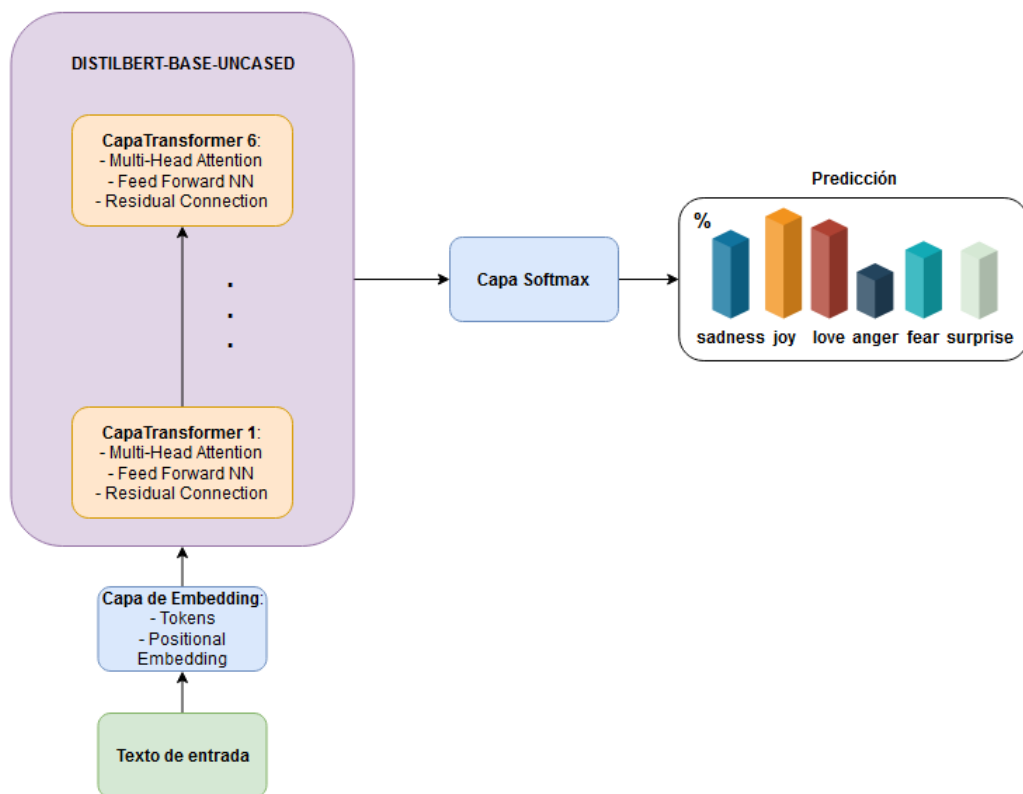
Además, la aplicación del modelo de clasificación de emociones a las reseñas de películas en el conjunto de datos proporciona un enfoque novedoso para el análisis crítico y la comprensión de las percepciones de las personas que han visto previamente una película del conjunto de datos. Al analizar las emociones presentes en las reseñas, se puede obtener una comprensión más profunda de cómo diferentes películas afectan y resuenan en las personas de manera emocional. Esto no solo enriquece la comprensión de las preferencias y gustos de los usuarios, sino que también puede ser utilizado para mejorar la recomendación de películas al considerar no solo géneros o tramas, sino también las emociones que una película evoca en su audiencia.

Para abordar la tarea de clasificación emocional en las respuestas de los usuarios y las reseñas de películas, se optó por aprovechar la potencia de los modelos pre entrenados en NLP. Uno de estos modelos, en particular, el "*distilbert-base-uncased*", se presenta como una elección

adecuada debido a su eficiencia y rendimiento en tareas de clasificación textual (Sanh et al., 2019).

Al emplear el modelo "*distilbert-base-uncased*", previamente entrenado en una amplia gama de textos, se aprovecha su capacidad para comprender el contexto y la semántica del lenguaje. Este modelo tiene la capacidad de captar matices emocionales sutiles presentes en el lenguaje humano, lo que es crucial para el éxito de la tarea de clasificación emocional en este proyecto (ver Figura 11).

Figura 11. Modelo "*distilbert-base-uncased*" con *fine tuning* para clasificación de emociones.



Fuente: elaboración propia.

Por otro lado, el conjunto de datos de *Democratizing Artificial Intelligence Research* (DAIR) (Saravia et al., 2018) se presenta como una fuente de información, con más de 416.000 textos recopilados de *tweets* y categorizados en seis emociones básicas: tristeza, alegría, amor, ira, miedo y sorpresa. Este amplio conjunto de datos proporciona una variedad de expresiones emocionales, que permite entrenar y afinar (*fine tuning*) el modelo de clasificación con una

diversidad de ejemplos reales. Por lo anterior, este conjunto de datos es el más adecuado para el desarrollo del *chatbot*.

El proceso de entrenamiento del modelo implica presentarle ejemplos del conjunto de datos de DAIR, donde cada texto está etiquetado con una de las seis emociones básicas. El modelo ajusta sus pesos y parámetros para aprender a asociar patrones lingüísticos específicos con cada emoción, como se muestra en la Figura 12. Una vez que el modelo está entrenado, se le puede utilizar para clasificar nuevas respuestas de usuarios y reseñas de películas en las emociones correspondientes.

Figura 12. Tweets presentes en el conjunto de datos DAIR con sus emociones.

	text	label	label_name
0	i can feel the end coming here i am amazed i m...	5	surprise
1	i guess i do feel the need to mention the real...	0	sadness
2	i have not read fiske so he may be right but i...	4	fear
3	i feel so awful	0	sadness
4	i bought a round it was also because i have no...	1	joy
...
333442	i am feeling so relieved that i was able to re...	1	joy
333443	i feel very positive toward this turn of events	1	joy
333444	i am feeling aggravated	3	anger
333445	i feel that the cold seafood here oysters lobs...	3	anger
333446	i do not feel burdened with the obligation to ...	0	sadness

333447 rows × 3 columns

Fuente: Saravia et al. (2018).

5.2.1. Entrenamiento del modelo

Para el proceso de entrenamiento, se hace uso de *Google Colab*, teniendo en cuenta que se requiere capacidad de procesamiento en GPU para esta tarea y esta plataforma permite el uso de capacidad de cómputo gratuito con hasta 16 GB de memoria VRAM.

Se crea un *notebook* de *Jupyter* en *Colab* y se importan las librerías *torch*, *transformers*, *datasets*, *numpy*, *pandas*, *matplotlib* y *scikit-learn*.

Figura 13. Importación del conjunto de datos para entrenamiento del modelo de clasificación de emociones.

```
1 # Cargar el dataset de DAR.AI:
2 emotions = load_dataset('json', data_files='data.jsonl', split='train')

1 # Dividir el dataset en conjunto de entrenamiento y test:
2 emotions = emotions.train_test_split(test_size=0.2)
3 emotions

DatasetDict({
  train: Dataset({
    features: ['text', 'label'],
    num_rows: 333447
  })
  test: Dataset({
    features: ['text', 'label'],
    num_rows: 83362
  })
})
```

Fuente: elaboración propia.

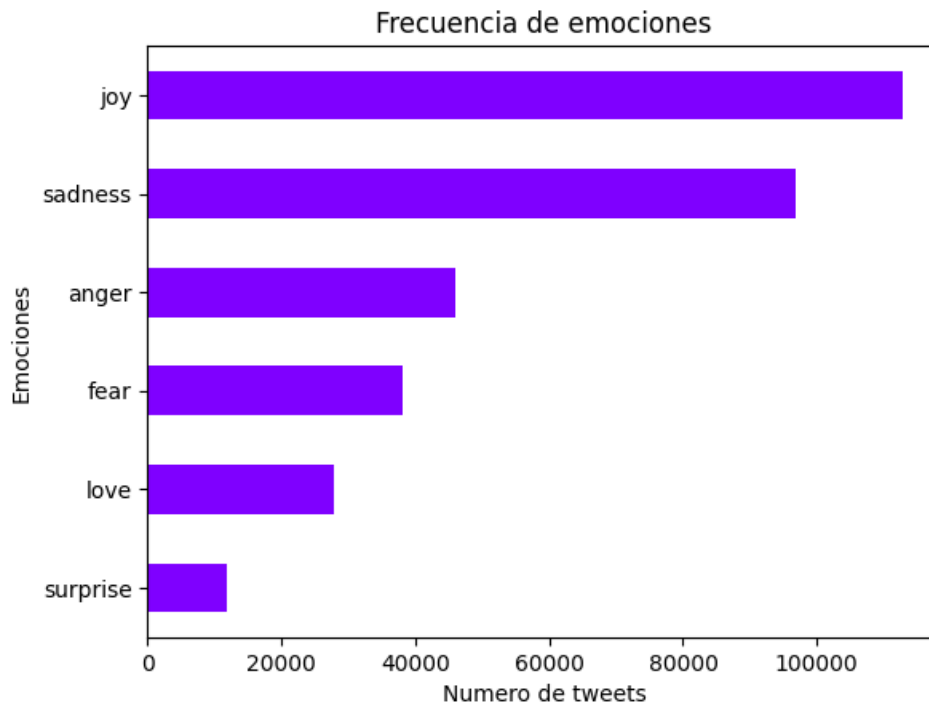
Se descarga del repositorio de *Hugging Face* de DAIR el conjunto de datos completo con los 416.809 *tweets* en formato *JSON* y se importa como un conjunto de datos de *Hugging Face* con la función *load_dataset()*. Posteriormente, se divide el conjunto de datos en conjunto de entrenamiento y prueba con una razón de 80/20 (ver Figura 13)

Analizando el conjunto de datos, se puede agrupar por emociones, la cantidad de *tweets* que entran en cada categoría. Se puede observar que las categorías alegría y tristeza son las que más aparecen en el conjunto de datos, seguidas de rabia, temor, amor y sorpresa. Esto tiene sentido si se tiene en cuenta la naturaleza de los textos que se encuentran comúnmente en los *tweets* (ver Figura 14).

También es importante conocer la distribución de la longitud textual de los *tweets*, no solo para comprender la estructura intrínseca de los datos, sino también para su adecuada preparación y adaptación al modelo *distilbert*. Dado que los modelos basados en *transformers*, tienen restricciones en cuanto a la longitud máxima de las secuencias de entrada, es relevante analizar la longitud de los *tweets* para asegurar una tokenización efectiva y evitar truncamientos excesivos que conllevan a pérdida de información y contexto, que puede afectar significativamente la precisión del modelo. En este caso, se puede apreciar en la Figura 15 que los *tweets* cuentan con una longitud promedio alrededor de 20 palabras, lo cual evita

tener que realizar truncamientos de texto, dado que el modelo *distilbert* permite un máximo de 512 *tokens* por cadena procesada.

Figura 14. Frecuencia de emociones presentes en el conjunto de datos de tweets.

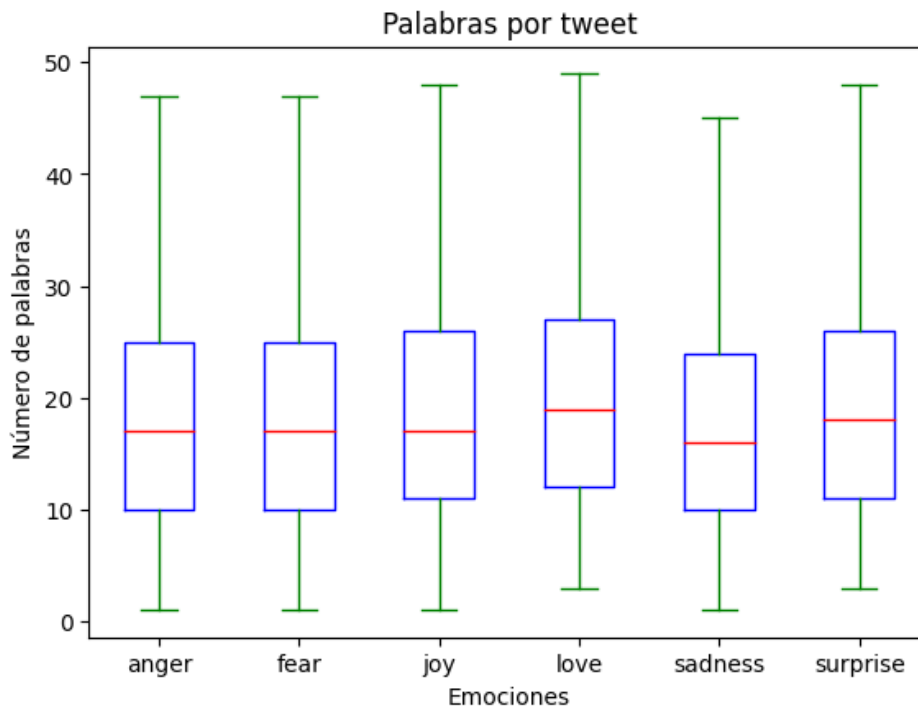


Fuente: elaboración propia.

Una vez hecho el anterior análisis, se procede a realizar la tokenización de cada *tweet* del conjunto de datos, requerida para poder hacer uso del modelo de *transformers distilbert-base-uncased*. Para esto, se usa la función *AutoTokenizer* de la librería *transformers* y haciendo uso del mismo modelo *distilbert*, como se muestra en la Figura 16.

El resultado de este proceso genera las columnas “*input_ids*” y “*attention_mask*” al conjunto de datos, que representan el vector de *embedding* y la máscara de atención como se muestra en la Figura 17. Cabe resaltar que después del proceso de tokenización se generó un vocabulario de *tokens* únicos de 30.522 *tokens* y al usar el *padding*, cada vector tiene un tamaño de 512 *tokens*.

Figura 15. Numero de palabras por tweet presentes en el conjunto de datos.



Fuente: elaboración propia.

Figura 16. Tokenización del conjunto de datos de tweets.

```
"""# Tokenización"""  
# Cargar el tokenizador:  
model_ckpt = "distilbert-base-uncased"  
tokenizer = AutoTokenizer.from_pretrained(model_ckpt)  
# Crear función para tokenizar el dataset:  
def tokenize(batch):  
    return tokenizer(batch["text"], padding=True, truncation=True)  
# Tokenizar el dataset:  
emotions_encoded = emotions.map(tokenize, batched=True, batch_size=None)
```

Fuente: elaboración propia.

Para el entrenamiento del modelo teniendo los *embeddings* y las máscaras de atención, primero se accede a la plataforma *Hugging Face* mediante una llamada a la función *notebook_login()*, que permite utilizar los recursos de la plataforma para el entrenamiento del modelo.

A continuación, se carga el modelo de clasificación utilizando la clase *AutoModelForSequenceClassification* de la librería. Se especifica el número de etiquetas de clasificación (*num_labels*) que en este caso son las 6 emociones y se envía el modelo a la unidad de procesamiento gráfico (GPU, por sus siglas en inglés) de *Colab*.

Figura 17. Muestra de *embedding* y máscara de atención para un tweet del conjunto de datos.

```
{'text': 'i feel liked ive been waiting for this book forever',
 'label': 2, 'input_ids': [101, 1045, 2514, 4669, 4921, 2063, 2042,
 3403, 2005, 2023, 2338, 5091, 102, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
 0, 0, 0, 0], 'attention_mask': [1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]}
```

Fuente: elaboración propia.

Se configuran los parámetros de entrenamiento, como el tamaño del lote (*batch_size*), la frecuencia de registro de métricas (*logging_steps*), el número de epochs de entrenamiento (*num_train_epochs*), la tasa de aprendizaje (*learning_rate*), entre otros. Estos parámetros se utilizan para definir un objeto *TrainingArguments* que encapsula la configuración del entrenamiento. Los parámetros más relevantes de configuración del modelo se presentan en la Tabla 5.

Tabla 5. Parámetros de entrenamiento del modelo de clasificación de emociones.

Parámetro	Valor	Descripción
<i>model_name</i>	<i>distilbert-base-uncased-finetuned-417k</i>	Nombre del modelo para ser usado posteriormente.
<i>batch_size</i>	64	Número de ejemplos de entrenamiento utilizados en cada iteración. Un tamaño de lote más grande puede acelerar el entrenamiento, ya que procesa más datos a la vez, pero también requiere más memoria. Al ser entrenado en <i>Colab</i> , 64 es el máximo que permite la cuenta gratuita, dada las limitaciones de memoria.
<i>num_train_epochs</i>	2	Número de veces que el modelo pasará por todo el conjunto de datos de entrenamiento. Un

		número adecuado de <i>epochs</i> puede prevenir el <i>overfitting</i> .
<i>learning_rate</i>	2.5×10^{-5}	Velocidad a la que el modelo se ajusta en función del error. Una tasa de aprendizaje demasiado alta puede causar que el modelo no converja, mientras que una tasa demasiado baja puede ralentizar el entrenamiento.
<i>weight_decay</i>	0.01	Técnica de regularización que ayuda a prevenir el <i>overfitting</i> al añadir una penalización a los pesos durante el entrenamiento.

Fuente: elaboración propia.

Posteriormente, se crea un objeto *Trainer* que se encarga de realizar el entrenamiento del modelo. Se le proporciona el modelo, los argumentos de entrenamiento, la función para calcular las métricas, los conjuntos de datos de entrenamiento y evaluación y el tokenizador necesario para procesar los datos de entrada.

Una vez configurado el entrenador, se inicia el entrenamiento mediante una llamada a la función *trainer.train()*. El modelo se entrena durante el número de *epochs* especificado en los argumentos de entrenamiento.

Finalmente, se realiza una evaluación del modelo utilizando los datos de prueba. Se utiliza la función *trainer.predict()* para obtener las predicciones del modelo sobre los datos de prueba y se almacenan en la variable *preds_output*.

La secuencia de pasos anteriormente descrita se puede ver en Figura 18.

Figura 18. Entrenamiento del modelo clasificador de emociones.

```

"""# Entrenar el modelo"""
# Acceder a Hugging Face:
notebook_login()
# Cargando el modelo de auto clasificación:
num_labels = 6
device = torch.device("cuda" if torch.cuda.is_available() else "cpu")
model = (AutoModelForSequenceClassification
         .from_pretrained(model_ckpt, num_labels=num_labels)
         .to(device))
# Configuración del entrenamiento:
batch_size = 64
logging_steps = len(emotions_encoded["train"]) // batch_size
model_name = f"{model_ckpt}-finetuned-emotion"
training_args = TrainingArguments(output_dir=model_name,
                                  num_train_epochs=2,
                                  learning_rate=2e-5,
                                  per_device_train_batch_size=batch_size,
                                  per_device_eval_batch_size=batch_size,
                                  weight_decay=0.01,
                                  evaluation_strategy="epoch",
                                  disable_tqdm=False,
                                  logging_steps=logging_steps,
                                  push_to_hub=True,
                                  log_level="error")

# Entrenando el modelo:
trainer = Trainer(model=model, args=training_args,
                  compute_metrics=compute_metrics,
                  train_dataset=emotions_encoded["train"],
                  eval_dataset=emotions_encoded["test"],
                  tokenizer=tokenizer)
trainer.train();

```

Fuente: elaboración propia.

Figura 19. Proceso de entrenamiento del modelo clasificador de emociones.

[10422/10422 44:00, Epoch 2/2]

Epoch	Training Loss	Validation Loss	Accuracy	F1
1	0.145800	0.090900	0.940884	0.939026
2	0.086000	0.088430	0.939241	0.938598

```

1 # Probar datos de entrenamiento:
2 preds_output_train = trainer.predict(emotions_encoded["train"])

1 # Métricas de entrenamiento:
2 preds_output_train.metrics

{'test_loss': 0.08020555227994919,
 'test_accuracy': 0.9448668004210564,
 'test_f1': 0.9426358051181883,
 'test_runtime': 1118.6361,
 'test_samples_per_second': 298.084,
 'test_steps_per_second': 4.658}

```

Fuente: elaboración propia.

El proceso de entrenamiento tuvo una duración de 44 minutos en 2 épocas y se obtuvieron las métricas presentes en la Figura 19 y Tabla 6.

Tabla 6. Resultado de métricas del modelo clasificador de emociones.

Dataset	Training Loss	Test Loss	Accuracy	F1 Score
Train	0.0802	-	0.9448	0.9426
Test	-	0.0843	0.9392	0.9385

Fuente: elaboración propia.

Para la evaluación de modelo de clasificación, se utiliza también la matriz de confusión que permite visualizar el desempeño del modelo para cada clase específica. Para esto, se crea una función *plot_confusion_matrix()*, que toma como argumentos las emociones reales, las emociones predichas y las etiquetas de las emociones, como se muestra en la Figura 20. Creación de la matriz de confusión para evaluar la precisión del modelo..

Figura 20. Creación de la matriz de confusión para evaluar la precisión del modelo.

```
"""# Creación de matriz de confusión"""
# crear matriz de confusión:
def plot_confusion_matrix(y_preds, y_true, labels):
    cm = confusion_matrix(y_true, y_preds, normalize="true")
    fig, ax = plt.subplots(figsize=(6, 6))
    disp = ConfusionMatrixDisplay(confusion_matrix=cm, display_labels=labels)
    disp.plot(cmap="BuGn", values_format=".2f", ax=ax, colorbar=False)
    plt.title("Matriz de confusión normalizada")
    plt.xlabel("Emociones predichas")
    plt.ylabel("Emociones verdaderas")
    plt.show()
# Crear las predicciones:
y_preds = np.argmax(preds_output.predictions, axis=1)
# Traer las etiquetas del conjunto de validación:
labels = ['sadness', 'joy', 'love', 'anger', 'fear', 'surprise']
y_valid = np.array(emotions_encoded["test"]["label"])
# Mostrar la matriz de confusión:
plot_confusion_matrix(y_preds, y_valid, labels)
```

Fuente: elaboración propia.

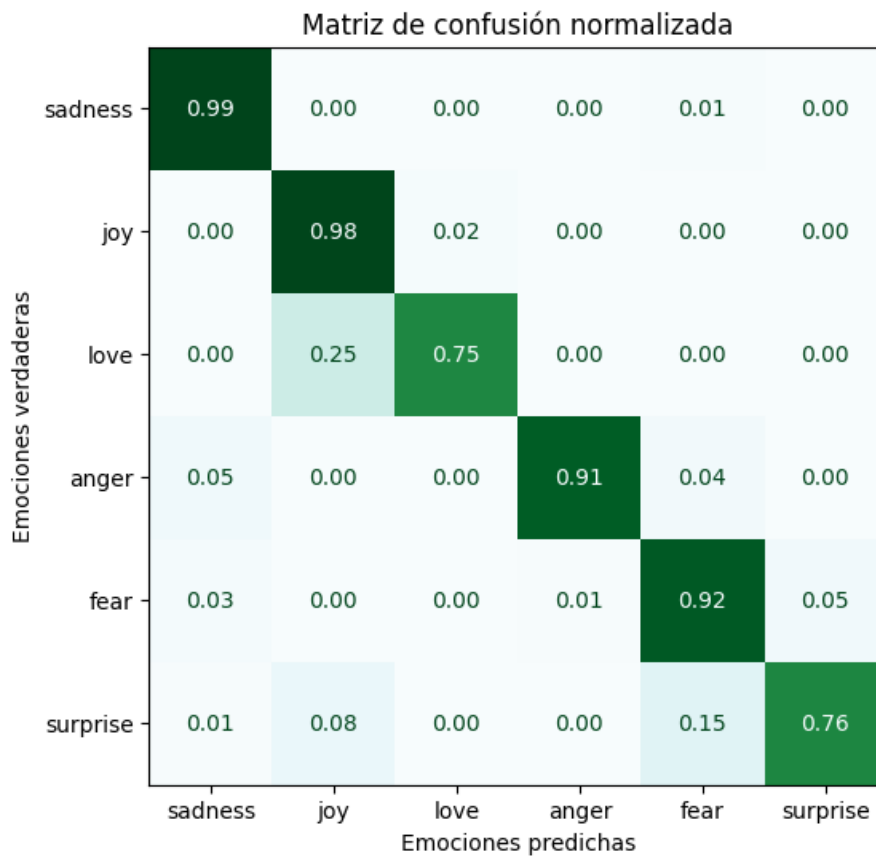
En la Figura 21, se muestra la matriz de confusión del modelo. Las filas de la matriz representan las emociones reales, mientras que las columnas reflejan las predicciones del modelo. Se puede observar que la matriz de confusión muestra que el modelo es más preciso a la hora de detectar tristeza (99%), seguida de alegría (98%). El modelo es menos preciso a la hora de

detectar la ira, el miedo, la sorpresa y el amor. También hay algunos falsos positivos y falsos negativos en la matriz de confusión.

Estos resultados sugieren que el modelo está funcionando adecuadamente en la clasificación de emociones, pero todavía hay margen de mejora, especialmente en la detección de emociones menos comunes como la sorpresa y el amor. Trabajos futuros podrían centrarse en mejorar la precisión del modelo a la hora de detectar estas emociones.

Una vez entrenado el modelo de clasificación de emociones, es crucial garantizar su disponibilidad para su uso en el *chatbot*. Para ello, el modelo fue subido a la plataforma *Hugging Face* a través de la función *push_to_hub()* (ver Figura 21), que no solo proporciona almacenamiento para modelos entrenados, sino también una API fácil de usar para hacer inferencias.

Figura 21. Matriz de confusión para evaluar la precisión del modelo de clasificación de emociones.



Fuente: elaboración propia.

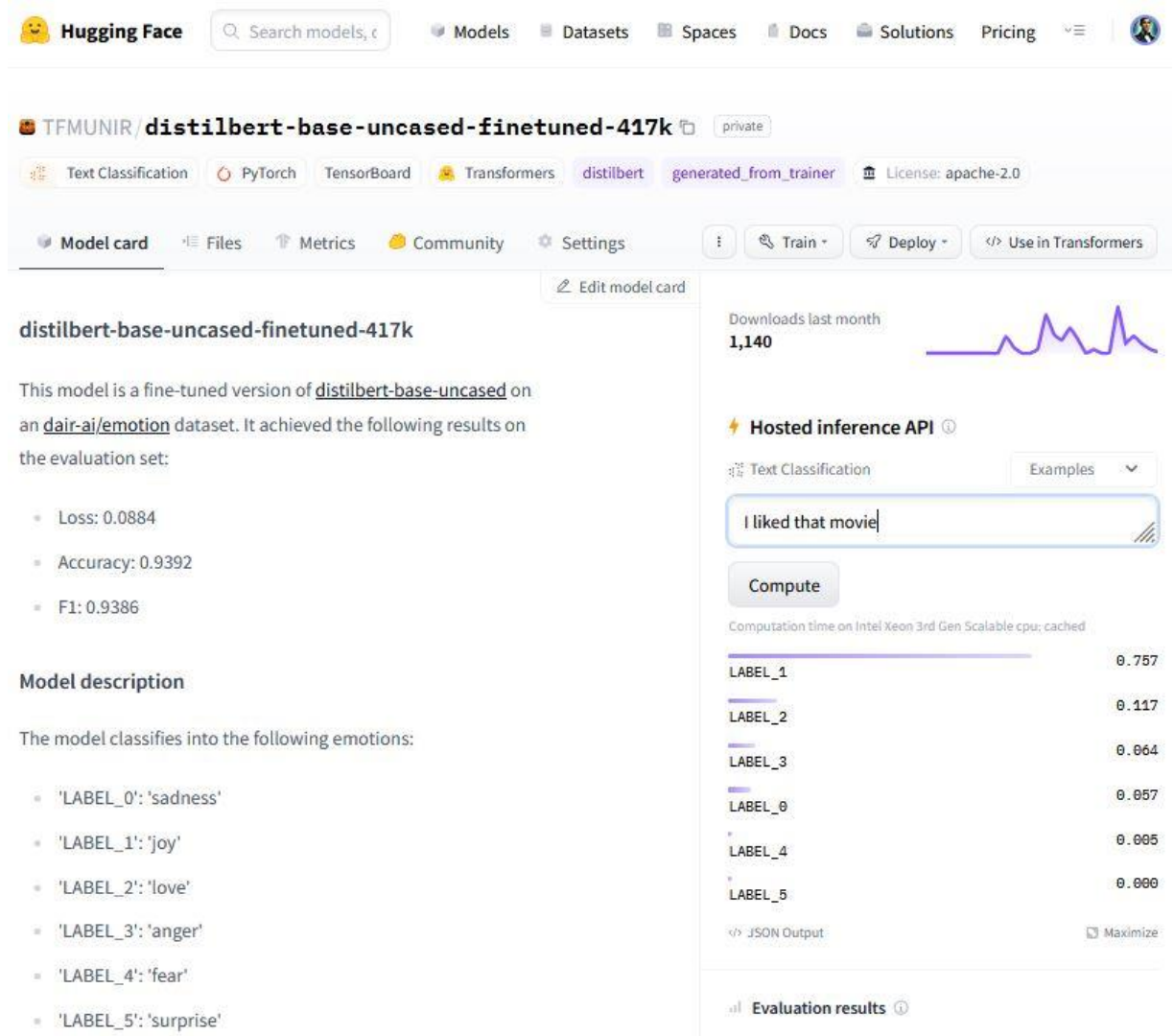
Figura 22. Creación de modelo de clasificación de emociones en la plataforma Hugging Face.

```
"""# Cargar el modelo en Hugging Face"""  
# subir el modelo a Hugging Face:  
trainer.push_to_hub(commit_message="Modelo de clasificación de emociones")
```

Fuente: elaboración propia.

El modelo fue subido a un repositorio de organización creado para este trabajo llamado: *TFMUNIR/distilbert-base-uncased-finetuned-417k* (ver Figura 23).

Figura 23. Modelo de clasificación de emociones cargado en la plataforma Hugging Face.



Fuente: elaboración propia.

5.2.2. Aplicación del modelo de clasificación al conjunto de datos de películas

Para realizar el análisis de emociones sobre el modelo entrenado previamente se elabora un *script* en *Python* para enviar las *reseñas* tokenizadas y realizar un truncamiento de texto de manera que tenga un tamaño de 512 *tokens* y crear un *dataframe* con el vector de probabilidades de emociones que refleja cada una de las *reseñas* de acuerdo con el modelo entrenado, como se ve en la Figura 24. Posteriormente se convierte en un diccionario y se guarda en una columna del *dataframe*.

Figura 24. Función para aplicación del modelo de clasificación de emociones al conjunto de datos de películas.

```
"""# Clasificación de emociones de las reviews"""
# Acceder a Hugging Face:
notebook_login()
# Cargar el modelo de emociones de Hugging Face:
model_id = "TFMUNIR/distilbert-base-uncased-finetuned-417k"
classifier = pipeline("text-classification", model=model_id)
# Función para clasificar las reviews y guardar en una columna:
def analytics_sentiments(review):
    mapping = {'LABEL_0': 'sadness',
               'LABEL_1': 'joy',
               'LABEL_2': 'love',
               'LABEL_3': 'anger',
               'LABEL_4': 'fear',
               'LABEL_5': 'surprise'}
    if review != '' and review != None:
        preds = classifier(review[:512], return_all_scores=True)
        preds_df = pd.DataFrame(preds[0])
        preds_df['label_name'] = preds_df['label'].replace(mapping)
        # Convertir el DataFrame en un diccionario
        preds_diccionario = preds_df.to_dict()
        return preds_diccionario
```

Fuente: elaboración propia.

Luego se ejecuta la función recorriendo el *dataframe* para la columna de *reseñas* por medio de la función *progress_apply()*. Este proceso tuvo una duración aproximada de 10.5 horas, dado que se ejecutó en CPU localmente (ver Figura 25).

Figura 25. Aplicación del modelo de clasificación de emociones al conjunto de datos de películas.

```
tqdm.pandas()
df_review_tokenization['sentiments'] = df_review_tokenization['tokenization'].progress_apply(analytics_sentiments)
```

0% | 1259/932464 [00:51<10:43:49, 24.11it/s]

Fuente: elaboración propia.

Este análisis de emociones se guarda como se ve en la Figura 26 tomando una fila como ejemplo.

Figura 26. Dataframe de películas con la columna de clasificación de emociones.

review	name	year	movie Rated	genres	tokenization	emotions
Confidently directed, dark, brooding, and packed with impressive action sequences and a complex story, The Dark Knight includes a career-defining turn from Heath Ledger as well as other Oscar worthy performances. TDK remains not only the best Batman movie, but comic book movie ever created.	The Dark Knight	2008	PG-13	Action; Crime; Drama;	confidently direct , dark , brooding , and pack with impressive action sequence and a complex story , the Dark Knight include a career - define turn from Heath Ledger as well as other Oscar worthy performance , TDK remain not only the good Batman movie , but comic book movie ever create	{ "label": {0: 'LABEL_0', 1: 'LABEL_1', 2: 'LABEL_2', 3: 'LABEL_3', 4: 'LABEL_4', 5: 'LABEL_5'}, "score": {0: 0.00016041890194173902, 1: 0.9994179010391235, 2: 1.1960293704760261e-05, 3: 0.00021328475850168816, 4: 0.00018620440096128732, 5: 1.037237325363094e-05}, "label_name": {0: 'sadness', 1: 'joy', 2: 'love', 3: 'anger', 4: 'fear', 5: 'surprise'} }

Fuente: elaboración propia.

Debido a que el análisis de emociones se realizó a todos los registros por *review*, se calculó un promedio de las reseñas por película de la variable *emotions* y así contar con un registro único por película de las emociones presentes en las *reseñas*, como se ve en la Figura 27.

En Figura 28, se muestra cómo se obtienen un total de 1.150 registros de emociones en el *dataframe*.

Finalmente, teniendo un conjunto de datos más limpio y tratado, se realizó una conexión con la API de TMDb para adicionar los nombres de las películas en latino, en castellano y la sinopsis de cada película en español latinoamericano, como se ve en la Figura 29.

Figura 27. Función para aplicar promedio de emociones de las reseñas por película.

```
"""# Promedio de emociones de las reviews"""
# Función para calcular el promedio de emociones:
def calculate_emotions_average(emotions_list):
    num_emotions = len(emotions_list)
    emotions_average = {
        'label': {},
        'score': {},
        'label_name': {}
    }
    for emotions_str in emotions_list:
        emotions_dict = ast.literal_eval(emotions_str)
        for key, value in emotions_dict['score'].items():
            emotions_average['score'][key] = emotions_average['score'].get(key, 0) + value / num_emotions
        for key, value in emotions_dict['label'].items():
            emotions_average['label'][key] = value
        for key, value in emotions_dict['label_name'].items():
            emotions_average['label_name'][key] = value
    return emotions_average
# Agregar las emociones promedio al dataset de películas:
average_emotions = selected_columns.groupby(['name',
                                             'year',
                                             'movie Rated',
                                             'genres'])['emotions'].agg(calculate_emotions_average).reset_index()
```

Fuente: elaboración propia.

Figura 28. Dataframe con promedio de emociones de las reseñas por película.

	name	year	movie Rated	genres	emotions
	10 Cloverfield Lane	2016	PG-13	Drama; Horror; Mystery;	{'label': (0: 'LABEL_0', 1: 'LABEL_1', 2: 'LABEL_2', 3: 'LABEL_3', 4: 'LABEL_4', 5: 'LABEL_5'), 'score': (0: 0.20293105520236593, 1: 0.44615120932427, 2: 0.00429838007514051, 3: 0.15658885566052452, 4: 0.1528144639756807, 5: 0.0372160309317203), 'label_name': (0: 'sadness', 1: 'joy', 2: 'love', 3: 'anger', 4: 'fear', 5: 'surprise')}
	10 Things I Hate About You	1999	PG-13	Comedy; Romance;	{'label': (0: 'LABEL_0', 1: 'LABEL_1', 2: 'LABEL_2', 3: 'LABEL_3', 4: 'LABEL_4', 5: 'LABEL_5'), 'score': (0: 0.088431959321473, 1: 0.6194725160723544, 2: 0.02277525540727719, 3: 0.18658373811725432, 4: 0.020264766690284004, 5: 0.06247176461900575), 'label_name': (0: 'sadness', 1: 'joy', 2: 'love', 3: 'anger', 4: 'fear', 5: 'surprise')}
	12 Angry Men	1957	Approved	Crime; Drama;	{'label': (0: 'LABEL_0', 1: 'LABEL_1', 2: 'LABEL_2', 3: 'LABEL_3', 4: 'LABEL_4', 5: 'LABEL_5'), 'score': (0: 0.12115408945735252, 1: 0.45931494089773406, 2: 0.008054366461066793, 3: 0.3558446508471902, 4: 0.02996652287285064, 5: 0.0256654303156497), 'label_name': (0: 'sadness', 1: 'joy', 2: 'love', 3: 'anger', 4: 'fear', 5: 'surprise')}

Fuente: elaboración propia.

Figura 29. Dataframe de películas con análisis de emociones y sinopsis.

	name	year	movie Rated	genres	emotions	name_latino	name_span	synopsis
0	10 Cloverfield Lane	2016	PG-13	Drama; Horror; Mystery;	{'label': (0: 'LABEL_0', 1: 'LABEL_1', 2: 'LABEL_2', 3: 'LABEL_3', 4: 'LABEL_4', 5: 'LABEL_5'), 'score': (0: 0.20293105520236593, 1: 0.44615120932427, 2: 0.00429838007514051, 3: 0.15658885566052452, 4: 0.1528144639756807, 5: 0.0372160309317203), 'label_name': (0: 'sadness', 1: 'joy', 2: 'love', 3: 'anger', 4: 'fear', 5: 'surprise')}	Avenida Cloverfield 10	Calle Cloverfield 10	Una joven sufre un accidente de coche. Cuando despierta se encuentra encerrada en una celda bajo tierra, secuestrada por un hombre extraño que dice haberla salvado del día del juicio final. El excéntrico captor asegura que el exterior es inhabitable por culpa de un terrible ataque químico, algo que no sabe si creer. Es así como comienza su pesadilla, en una espiral de desconocimiento. La mente del criminal es infranqueable, y ella tendrá que intentar analizar a su enemigo para conseguir sobrevivir.
1	10 Things I Hate About You	1999	PG-13	Comedy; Romance;	{'label': (0: 'LABEL_0', 1: 'LABEL_1', 2: 'LABEL_2', 3: 'LABEL_3', 4: 'LABEL_4', 5: 'LABEL_5'), 'score': (0: 0.088431959321473, 1: 0.6194725160723544, 2: 0.02277525540727719, 3: 0.18658373811725432, 4: 0.020264766690284004, 5: 0.06247176461900575), 'label_name': (0: 'sadness', 1: 'joy', 2: 'love', 3: 'anger', 4: 'fear', 5: 'surprise')}	10 cosas que odio de ti	10 razones para odiarte	Mientras que por un lado Bianca es joven, bella, popular y egoísta, Kat es la chica rebelde y feminista que está en contra de todo y de todos, que gusta de leer literatura que para muchos sería subversiva y que suele vestirse de manera un tanto estrafalaria. Su padre Walter es un hombre de una estricta educación y tiene una regla

Fuente: elaboración propia.

5.2.3. Segmentación del conjunto de datos de películas

Se decidió realizar una segmentación del conjunto de datos debido a la necesidad de incrementar la variabilidad y distribución en las recomendaciones, garantizando un enfoque más dirigido y personalizado. Al dividir el conjunto de datos en grupos homogéneos basados en criterios como género, año o emociones, se posibilita una personalización más afinada de las recomendaciones, alineándolas con las emociones del usuario en un momento específico.

Esta estrategia no solo permite la diversificación de las sugerencias ofrecidas, sino que también garantiza que no se concentren exclusivamente en un tipo o emoción particular. Adicionalmente, la segmentación puede facilitar el descubrimiento de contenido, resaltando películas o géneros que podrían ser inadvertidos en sistemas de recomendación convencionales. Desde el punto de vista técnico, segmentar el conjunto de datos podría optimizar el rendimiento del sistema de recomendación, ya que no requeriría el análisis completo del conjunto de datos en cada interacción. Este enfoque también mejora la interpretabilidad del modelo, ya que las recomendaciones basadas en segmentos específicos ofrecen mayor claridad en términos de justificación.

Para lograr este objetivo, se realizaron las siguientes operaciones al conjunto de datos:

1. Eliminar las columnas de texto que no son relevantes para el análisis como: *name*, *name_latino*, *name_spain*; quedando en el conjunto de datos las columnas: *year*, *movie Rated*, *genres*. Las probabilidades de las emociones se colocaron en 6 columnas independientes.
2. Se codificó la columna *movie Rated* en cinco categorías: 0: todo público, 1: mayores de 8 años, 2: mayores de 14 años, 3: mayores de 17 años y 4: mayores de 18 años, alineado con el sistema de clasificación de *Motion Picture Association* (MPA).
3. Al analizar la columna correspondiente a los géneros (*genres*) del conjunto de datos, una película puede estar asociada con múltiples géneros simultáneamente. Esta complejidad refleja la diversidad temática que una película puede presentar. Para transformar adecuadamente esta información en un formato que pueda ser procesado y analizado de manera eficiente, se recurrió a la técnica de *One Hot Encoding*. Esta metodología consiste en convertir las categorías nominales en un formato binario, lo que resultó en la creación de 182 categorías de géneros distintos. Si bien, esta cifra

podría interpretarse como una dimensionalidad excesiva en términos de características, es fundamental considerar la naturaleza multitemática del cine.

4. Dentro del proceso de preparación y transformación de datos, se optó por codificar adecuadamente las sinopsis (*synopsis*, en inglés) de las películas, las cuales representan textos ricos en contenido semántico y narrativo. Para ello, se implementó de la técnica de *Term Frequency-inverse Document Frequency (TFIDF)*, por sus siglas en inglés).

Esta técnica, permite representar los textos basándose en la importancia relativa de cada término en el documento en relación con un conjunto de documentos. A través de *TFIDF*, se otorga un peso significativo a los términos que son frecuentes en una sinopsis específica pero no en la mayoría de las sinopsis, capturando así la singularidad y relevancia de palabras específicas para cada película.

Luego de la aplicación de esta técnica, se obtuvieron 1000 *features*, representando la diversidad y complejidad de las sinopsis. Posteriormente, para agrupar y sintetizar esta información, se aplicó una técnica de clusterización, resultando en la definición de 10 *clusters* distintos para las sinopsis.

Dada la alta dimensionalidad del conjunto de datos procesado, el resumen de todas estas operaciones se puede ver en la Figura 30.

Figura 30. Conjunto de datos codificado para ser usado en el análisis de segmentación.

```
1 df_cluster.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1150 entries, 0 to 1149
Columns: 192 entries, year to cluster_synopsis
dtypes: float64(6), int32(1), int64(2), uint8(183)
memory usage: 282.0 KB
```

Fuente: elaboración propia.

Tras la codificación del conjunto de datos y previo a la clusterización, se debe garantizar que la selección del número de agrupaciones (*clusters*, en inglés) se realice de manera óptima. Para ello, se emprendió un análisis basado en la variación de la distorsión en relación con la cantidad de *clusters*. La distorsión, se refiere a la suma de las distancias al cuadrado de cada punto a su centroide más cercano; Un menor valor de distorsión sugiere que los puntos están

más cerca de los centros de sus agrupaciones respectivas, y por lo tanto, el agrupamiento podría considerarse más cohesivo.

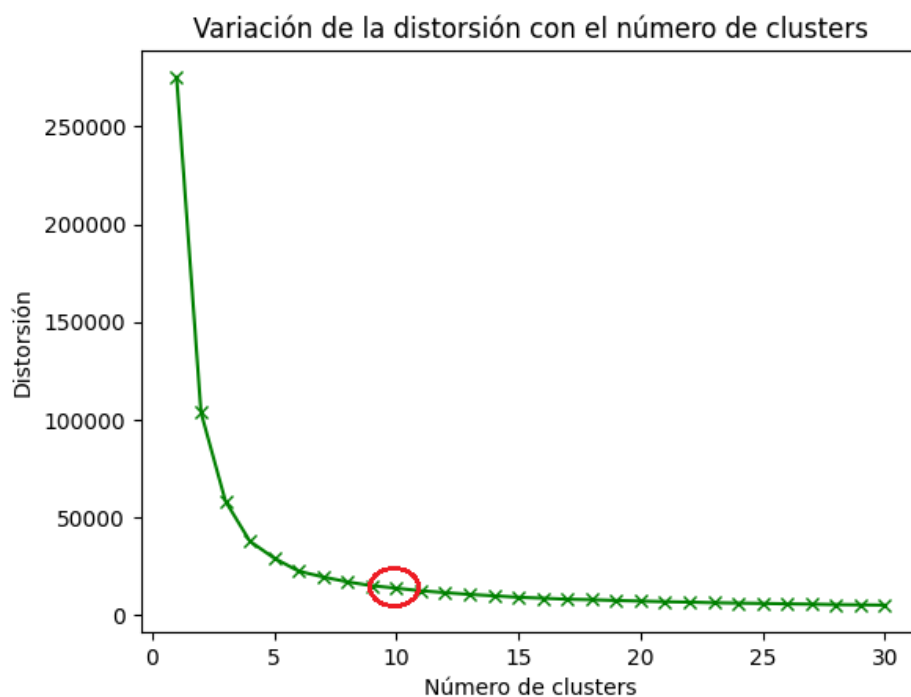
Sin embargo, es fundamental mencionar que existe una limitación al usar la distorsión como criterio. Al incrementar el número de agrupaciones, la distorsión tiende a disminuir. Esto no necesariamente implica una mejor segmentación, ya que un número excesivo de agrupaciones puede sobreajustar los datos y perder generalidad en las agrupaciones. Por lo tanto, es común recurrir al método del codo (*elbow method*, en inglés) en este análisis, buscando el punto en el cual el decremento de la distorsión comienza a atenuarse, indicando un equilibrio entre precisión y generalidad.

Con base en las consideraciones previas y de acuerdo con la Figura 31, se estableció la necesidad de analizar un espectro de segmentos que abarca desde 1 hasta 30 agrupaciones. A partir de esta exploración, se observó que un punto óptimo que proporciona una segmentación diversificada y representativa de las películas se sitúa alrededor de los 10 agrupaciones. Esta elección se alinea con un equilibrio entre la especificidad y la generalidad: se distancia del valor identificado por el método del codo, que sugiere $k = 5$, evitando una posible subsegmentación, pero a su vez, no se inclina hacia valores excesivamente elevados, superiores a $k = 10$, que podrían conllevar una sobresegmentación y por ende, reducir la robustez y utilidad práctica de las agrupaciones definidas.

Para $k = 10$, la segmentación de las películas se puede observar en la Figura 32.

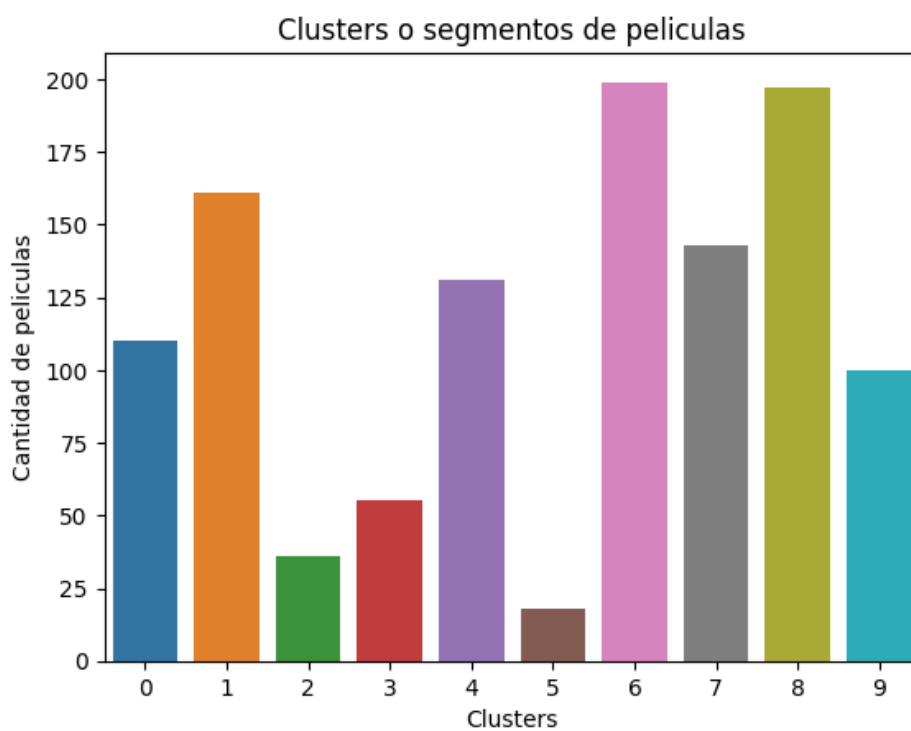
Finalmente, se elaboró un gráfico bidimensional, con el propósito de visualizar de manera clara y concisa la distribución de las películas en un espacio proyectado a través de un Análisis de Componentes Principales (PCA, por sus siglas en inglés), como se ve en la Figura 33. En esta se presenta la agrupación inherente de las películas basada en las características extraídas y de las agrupaciones formadas.

Figura 31. Variación de la distorsión con el número de agrupaciones o clusters.



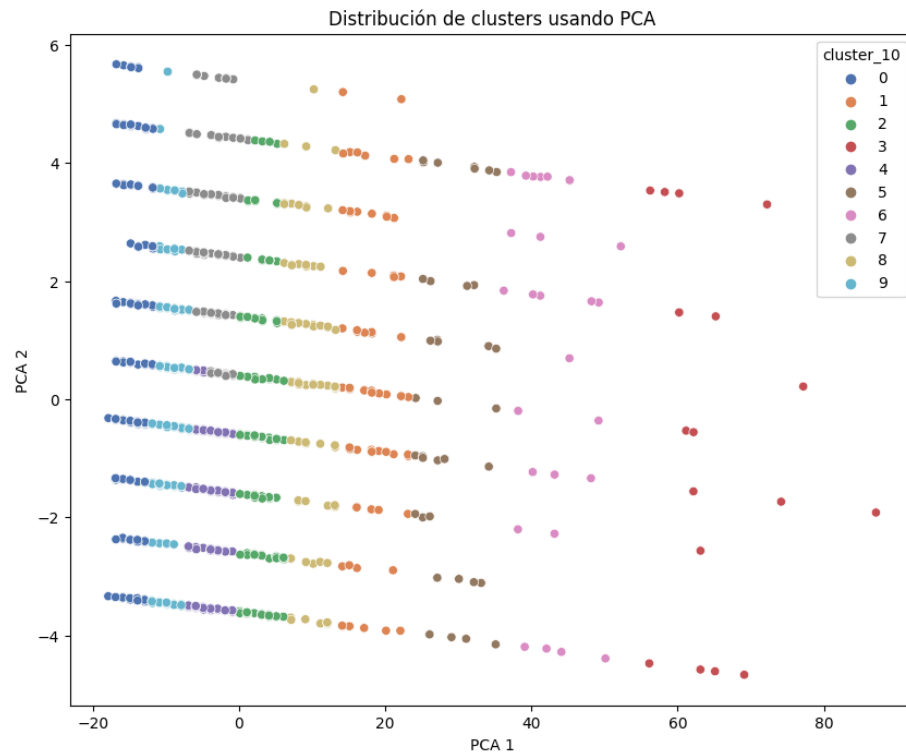
Fuente: elaboración propia.

Figura 32. Cantidad de películas en cada agrupación o clúster.



Fuente: elaboración propia.

Figura 33. Distribución de las diez agrupaciones usando dos componentes principales.



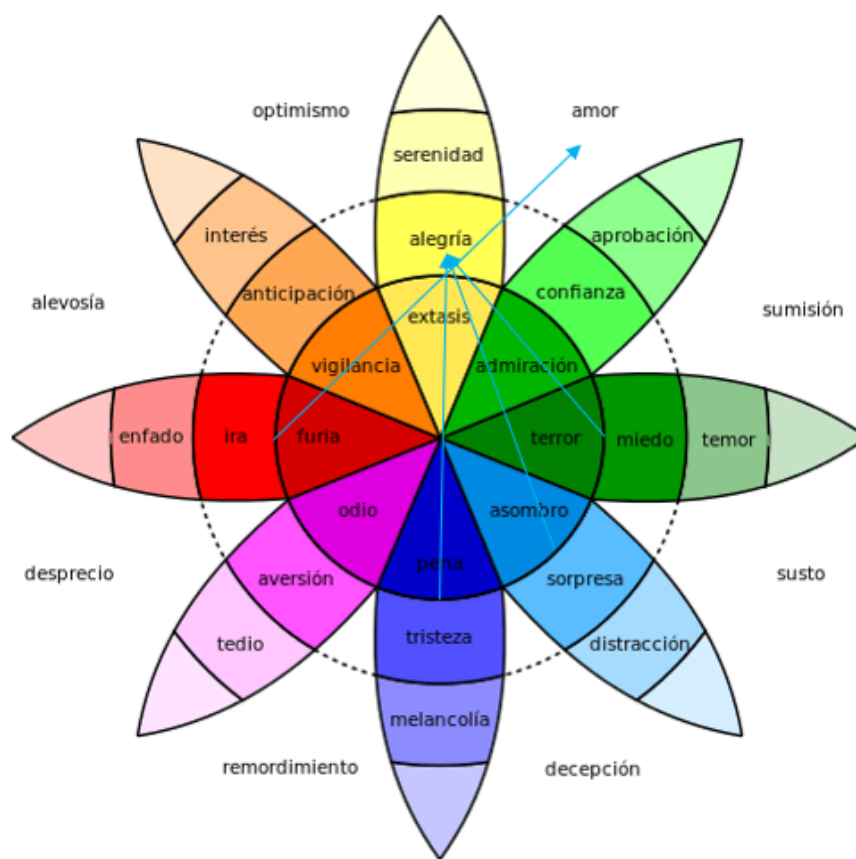
Fuente: elaboración propia.

5.2.4. Reajuste emocional para recomendación de películas

Con base en la rueda de las emociones presentada en la Sección 2, se desarrolló un modelo que reajusta emociones negativas detectadas en las interacciones del usuario con el *chatbot*, orientándolas hacia sus contrapartes positivas. El objetivo de este ajuste es que, al identificar emociones tales como el miedo, la tristeza o el enojo, estas sean procesadas por un algoritmo que, sin eliminar la emoción negativa original, pondere hacia su opuesto positivo.

Esta metodología tiene como finalidad ofrecer recomendaciones de películas que sean más amenas y placenteras, con el propósito de contribuir a la mejora del estado anímico del usuario. Esta estrategia se alinea con la idea de proporcionar una experiencia de usuario optimizada y empática, reconociendo y respondiendo adecuadamente a sus estados emocionales. La propuesta de balance de emociones se muestra en la Figura 34.

Figura 34. Contrapartes de emociones implementadas en el chatbot.



Emoción del usuario	Contraparte emocional del modelo
Tristeza (sadness)	Alegría (joy)
Alegría (joy)	Alegría (joy)
Amor (love)	Amor (love)
Rabia (anger)	Amor (love)
Temor (fear)	Alegría (joy)
Sorpresa (surprise)	Alegría (joy)

Fuente: elaboración propia.

Figura 35. Función de balance de emociones implementadas en el chatbot.

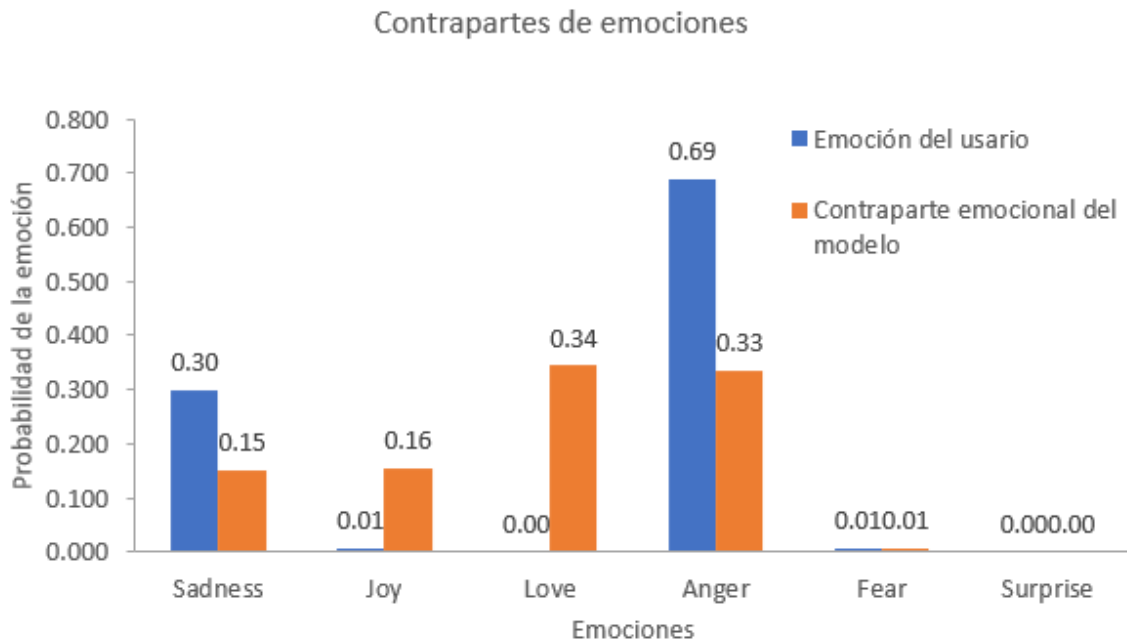
```
"""# Balance de emociones"""  
# Creación de la función balanceadora de emociones:  
def vector_balance(vector_usuario):  
    mapping = {0: 'sadness', 1: 'joy', 2: 'love', 3: 'anger', 4: 'fear', 5: 'surprise'}  
    # Contrapartes emocionales propuestas:  
    contrapartes = {  
        'sadness': 'joy',  
        'joy': 'joy',  
        'love': 'love',  
        'anger': 'love',  
        'fear': 'joy',  
        'surprise': 'joy'  
    }  
    vector_equilibrado = np.copy(vector_usuario)  
  
    factor_ajuste = 0.5  
  
    for i, valor in enumerate(vector_usuario):  
        emocion_actual = mapping[i]  
        contraparte = contrapartes[emocion_actual]  
  
        if valor > 0.2: # Si la emoción tiene un valor significativo:  
            indice_contraparte = [k for k, v in mapping.items() if v == contraparte][0]  
            vector_equilibrado[indice_contraparte] += valor * factor_ajuste # Reforzamos la contraparte  
            vector_equilibrado[i] -= valor * factor_ajuste # Atenuamos la emoción original  
  
    # Normalizar el vector equilibrado para que sume 1:  
    vector_equilibrado = vector_equilibrado / vector_equilibrado.sum()  
    return vector_equilibrado
```

Fuente: elaboración propia.

En la implementación del modelo, específicamente si una emoción como la tristeza, excede un umbral definido (en este caso 0.2), el modelo incrementa su contraparte positiva (alegría) mientras reduce proporcionalmente la emoción original. Esta operación se realiza mediante un factor de ajuste, que en este contexto se ha establecido en 0.5. Posteriormente, para garantizar una respuesta coherente, el vector resultante es normalizado, de modo que la suma total de sus componentes emocionales o probabilidades sea 1. De esta manera, el sistema proporciona una respuesta emocionalmente equilibrada y adaptada al contexto del usuario.

La función `vector_balance()`, fue implementada y forma parte del backend del chatbot, dado que depende de las interacciones del usuario en el frontend (ver Figura 35).

Figura 36. Ejemplo de funcionamiento del reajuste emocional en el chatbot.



Frase: “Un día muy agitado, demasiado trabajo no tengo tiempo de nada”

Fuente: elaboración propia.

Un ejemplo de cómo funciona el balance emocional del vector de emociones del usuario para la recomendación de películas se puede ver en la Figura 36. En respuesta a la pregunta: “Cuéntame de forma resumida, ¿Cómo ha transcurrido tu día hasta ahora y que has hecho?”, un usuario podría responder “Un día muy agitado, demasiado trabajo no tengo tiempo de nada”. El texto ingresa al modelo de clasificación de emociones, arrojando como resultado un vector de probabilidades de emociones $[0.300, 0.006, 0.000, 0.688, 0.006, 0.000]$ y su emoción balanceada calculada es $[0.150, 0.156, 0.344, 0.334, 0.006, 0.000]$. En este caso, las emociones más predominantes son *anger* y *sadness*, las cuales fueron atenuadas y sus contrapartes emocionales *love* y *joy* fueron acentuadas; además, las emociones que no están por encima del umbral establecido permanecen iguales.

5.3. Desarrollo del sistema de recomendación y chatbot

5.3.1. Descripción del sistema de recomendación propuesto

A continuación, se brinda una descripción del sistema de recomendación desarrollado bajo el enfoque de la taxonomía propuesta en la Sección 2 del presente trabajo. El objetivo es

delimitar adecuadamente los alcances del sistema y proporcionar una referencia clasificatoria en cuanto a las técnicas y métodos creados para el sistema más adelante.

1. **Dominio:** el sistema de recomendación propuesto se encuentra circunscrito al ámbito cinematográfico, específicamente, a la selección y recomendación de películas, el fin principal del sistema a crear es influir positivamente en el estado de ánimo de los usuarios. La precisión de las recomendaciones estará intrínsecamente relacionada con la cantidad y calidad del texto proporcionado por los usuarios como entrada. Asimismo, cabe destacar que la posibilidad de recomendar una película en repetidas ocasiones dependerá de la detección de emociones similares por parte del modelo clasificatorio, con la incorporación de un componente de aleatoriedad que permitirá variaciones en las recomendaciones.
2. **Propósito:** suministro de recomendaciones adecuadas y pertinentes a los estados de ánimo actuales de los usuarios. Para el proveedor, el objetivo fundamental es mejorar la calidad de las recomendaciones ofrecidas por los sistemas tradicionales, adaptándolas a los estados emocionales de los usuarios de manera más efectiva. Por otro lado, para el usuario, el propósito es acceder a sugerencias relevantes que, de otra manera, podrían resultar más difíciles de identificar dentro de la amplia variedad de opciones disponibles en los sistemas de recomendación comerciales.
3. **Contexto:** el sistema se enfoca exclusivamente en contenido textual proporcionado por el usuario para discernir su estado emocional y, en base a ello, generar recomendaciones relevantes. Este análisis se orienta a identificar las emociones más recientes experimentadas por el usuario, prescindiendo de elementos externos como dispositivos utilizados, ubicación geográfica, horario o actividades en curso.
4. **Nivel de personalización:** el nivel de personalización del sistema de recomendación propuesto es de carácter plenamente personalizado. En este contexto, cada usuario recibe recomendaciones individualizadas y exclusivas, las cuales se fundamentan en la detección de emociones y estados de ánimo expresados a través del contenido textual proporcionado. El sistema prescinde de la clasificación de recomendaciones por popularidad, en lugar de ello, se enfoca en ofrecer sugerencias de películas que exhiban similitudes emocionales con las expresadas por el usuario, garantizando así una experiencia personalizada y adaptada a sus necesidades y preferencias particulares.

5. **Privacidad y confiabilidad:** el sistema de recomendación propuesto dará un alto nivel de importancia a la protección de la privacidad de los usuarios. La recopilación de datos se realiza en dos fases: La primera para el registro del usuario, en la que se solicita información de nombre de usuario, nombre, apellido, correo electrónico y establecimiento de contraseña de acceso. Todo esto bajo un acuerdo de términos y condiciones y política de tratamiento de datos. La segunda, se limita exclusivamente a la obtención de información relacionada con la edad del usuario durante la conversación con el *chatbot*, y esto se realiza con el único propósito de cumplir con las restricciones de edad impuestas por las entidades de clasificación. Cabe destacar que en ningún momento se solicita ni se almacena información adicional de carácter personal, como género o estado civil, con el fin de salvaguardar la privacidad del usuario.

El sistema de recomendación no tiene la intención de influir en los textos proporcionados por el usuario como entrada. El algoritmo empleado se enfoca en generar recomendaciones basadas exclusivamente en el contenido textual suministrado por el usuario, evitando cualquier forma de manipulación o influencia.

La información recopilada se utiliza únicamente con propósitos de mejora y evaluación del sistema de recomendación, y no se comparte ni se divulga con terceros. Este enfoque en la privacidad y la integridad de los datos contribuye a establecer un alto nivel de confiabilidad en las sugerencias proporcionadas, asegurando que los usuarios perciban el sistema como una herramienta seria y objetiva en lugar de una plataforma con motivaciones comerciales o manipulativas.

6. **Interfaz:** la comunicación entre el sistema de recomendación y el usuario se realizará exclusivamente mediante una interfaz de texto escrito, por lo que el usuario deberá tener las habilidades necesarias para poder establecer:

- 6.1. **Entrada:** los usuarios tienen la capacidad de proporcionar respuestas, las cuales son utilizadas como información de entrada por el sistema. A partir de estas respuestas, el sistema deducirá las recomendaciones más adecuadas y pertinentes para cada usuario, con el objetivo de mejorar su estado emocional. La entrada de datos puede considerarse implícita, ya que el sistema utiliza las respuestas del usuario para inferir sus preferencias y estados emocionales, sin requerir una introducción explícita de información sobre gustos o preferencias por parte del consumidor.

6.2. Salida: la salida del sistema se traduce en recomendaciones personalizadas de películas diseñadas para influir positivamente en el estado emocional del usuario. Estas recomendaciones se presentan en forma de sugerencias específicas y adaptadas a las necesidades emocionales de cada usuario.

El sistema de recomendación se clasifica como un "recomendador de caja negra", debido a que no se proporciona una explicación detallada sobre los motivos detrás de las recomendaciones. Los usuarios no tendrán acceso a información sobre el proceso de generación de las sugerencias, debido a que el sistema opera de manera oculta en este aspecto, sin revelar los algoritmos o criterios específicos utilizados para generar las recomendaciones.

5.3.2. Chatbot: AngellA

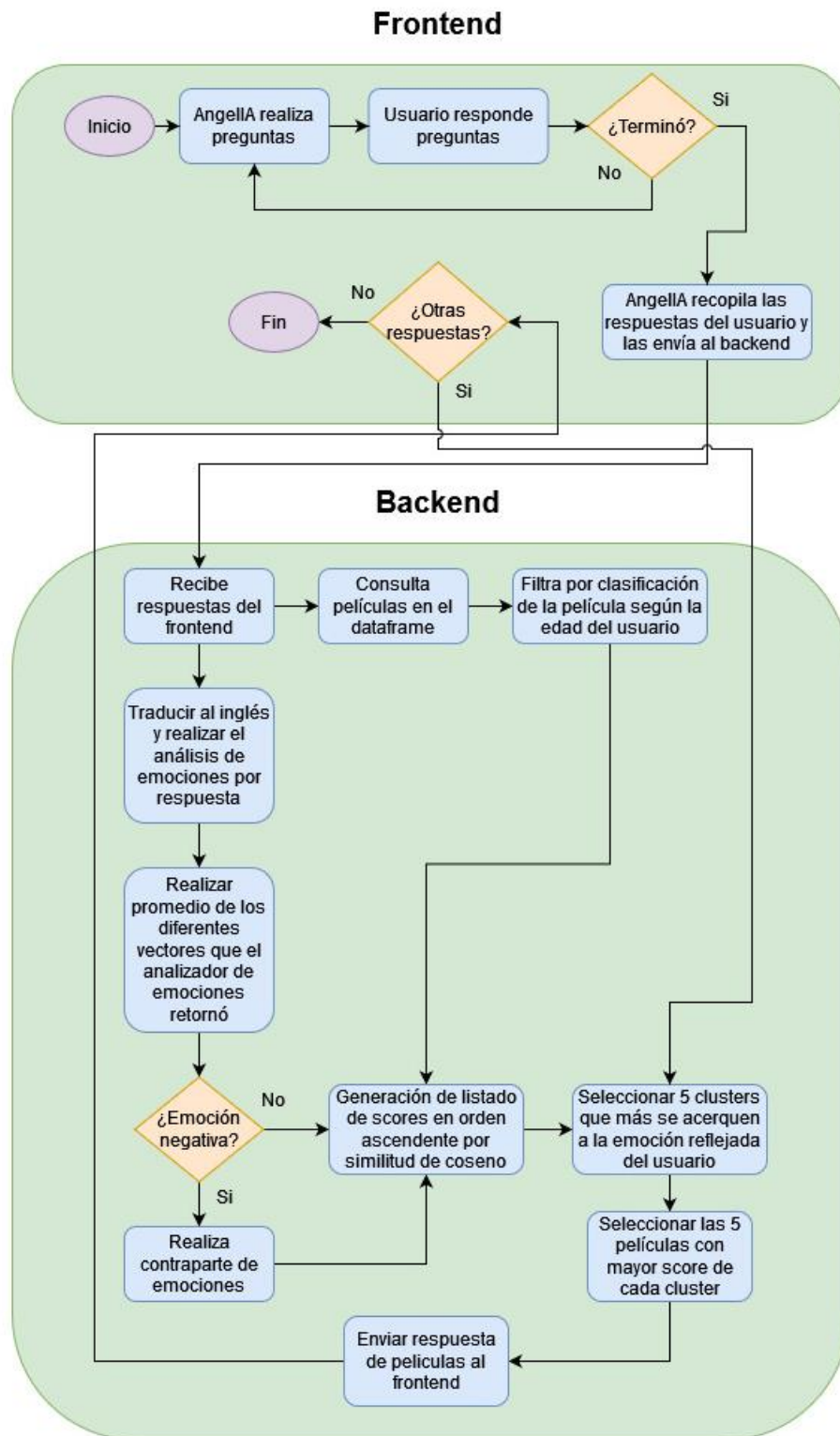
"AngellA" es un sistema de IA diseñado para brindar a los usuarios recomendaciones de películas personalizadas a través de interacciones en tiempo real. Para lograr esta funcionalidad, el sistema se apoya en un *backend* desarrollado en el *framework Django* (Django Software Foundation, 2023), que comprende una variedad de funciones y vistas para llevar a cabo tareas esenciales como procesamiento de texto, análisis de emociones, recomendación de películas e interacción con el usuario. Además, el sistema se completa con un *frontend* desarrollado en HTML (World Wide Web Consortium, 2023), JavaScript (ECMA International, 2023) y CSS (World Wide Web Consortium, 2017), que brinda la interfaz con la que los usuarios interactúan.

El diagrama de la Figura 37 presenta el funcionamiento simplificado de AngellA.

5.3.3. Elección del sistema de generación de texto del chatbot

En una primera etapa, se exploró la viabilidad de utilizar la plataforma de *Hugging Face* para implementar un sistema en la nube capaz de realizar inferencias con LLM en formato *Open Source*. Si bien esta plataforma ofrecía una amplia gama de modelos pre entrenados, surgió un desafío importante: los costos computacionales asociados al mantenimiento de dicho sistema en la nube eran significativamente elevados. Esto planteaba una preocupación en términos de escalabilidad y sostenibilidad económica.

Figura 37. Diagrama de flujo del funcionamiento simplificado de AngellA.



Fuente: elaboración propia.

Adicionalmente, se identificó un problema relacionado con la memoria. Algunos de los modelos cuantizados en español requerían un mínimo de 8 GB de RAM para funcionar de

manera adecuada. Esta limitación de recursos de hardware complicaba la implementación de un modelo LLM, especialmente en entornos con recursos limitados.

En un intento por explorar alternativas, se consideró la librería de *Python* conocida como *Chatterbot*. No obstante, se encontró con un obstáculo importante: *Chatterbot* carecía de soporte activo y actualizaciones, lo que dificultaba su adaptación a las últimas versiones de *Python* 3. La perspectiva de tener que adaptar todo el código a *Python* 2 aumentaba la complejidad de la tarea y planteaba interrogantes sobre la viabilidad de esta opción.

También se evaluó *Microsoft Virtual Agents* como una posible solución. Aunque esta plataforma ofrecía una capacidad de procesamiento sólida y recursos disponibles, se observó que su capacidad de personalización era limitada. El enfoque principal de *Microsoft Virtual Agents* se centraba en la atención al cliente y la automatización de respuestas, lo que no satisfacía por completo las necesidades de personalización requeridas para el sistema de recomendación específico.

Finalmente, se incorporó el sistema conversacional al *framework Django*. Dicho sistema de conversación opera a través de un *chatbot* fundamentado en un conjunto de reglas preestablecidas. Estas definen anticipadamente las interrogantes que serán planteadas por el *chatbot*, mientras que las respuestas proporcionadas por el usuario serán procesadas y analizadas por el sistema para generar recomendaciones efectivas.

5.3.4. Backend

El *backend* actúa como el cerebro del sistema, coordinando y ejecutando todas las operaciones y procesos necesarios para brindar recomendaciones de películas personalizadas. Este tiene los siguientes aspectos principales:

1. **Análisis de emociones:** uno de los aspectos clave de *AngellIA* es su capacidad para entender y analizar las emociones detrás de los mensajes de los usuarios se logra utilizando el modelo pre entrenado, que previamente ha sido cargado en *Hugging Face*. La función principal en esta etapa es *emotion_analysis()*. Esta función (ver Figura 38) toma el mensaje proporcionado por el usuario, lo traduce al inglés (para asegurar una consistencia en el análisis), y luego lo clasifica en categorías de emociones. El modelo pre entrenado es el encargado de realizar este análisis de emociones y generar un diccionario de emociones correspondiente a los mensajes de usuario.

2. **Procesamiento de texto:** la función *find_sentences_with_noun_verb()* (ver Figura 39) juega un papel esencial en el procesamiento del texto de los usuarios. Su función principal es identificar y extraer oraciones que contengan sustantivos, verbos o adjetivos. Para lograr esto, se utiliza la librería de *Python SpaCy* para NLP.

Figura 38. Función analizadora de emociones del backend.

```
"""# Análisis de emociones"""
# Creación de la función para analizar emociones:
def emotion_analysis(message):

    mapping = {'LABEL_0': 'sadness',
               'LABEL_1': 'joy',
               'LABEL_2': 'love',
               'LABEL_3': 'anger',
               'LABEL_4': 'fear',
               'LABEL_5': 'surprise'}

    # Crea una instancia del pipeline:
    model_id = "TFMUNIR/distilbert-base-uncased-finetuned-417k"
    access_token = "HUGGINGFACEHUB_API_TOKEN"
    nlp_classifier = pipeline("text-classification", model=model_id, use_auth_token=access_token)
    translated = str(GoogleTranslator(source='auto', target='en').translate(str(message)))

    preds = nlp_classifier(str(translated[:512]), return_all_scores=True)
    preds_df = pd.DataFrame(preds[0])

    preds_df['label_name'] = preds_df['label'].replace(mapping)
    # Convertir el DataFrame en un diccionario:
    preds_diccionario = preds_df.to_dict()
    return preds_diccionario
```

Fuente: elaboración propia.

Esta realiza un análisis lingüístico del texto de entrada para identificar los componentes clave del texto ingresado por el usuario. Esto ayuda a comprender la estructura gramatical y las relaciones entre las palabras en el mensaje. La detección de sustantivos, verbos y adjetivos es crucial para garantizar una coherencia gramatical del texto. Si la oración no es coherente, la función retorna un *false*, haciendo que el *chatbot* solicite nuevamente una entrada de texto para la misma pregunta.

Figura 39. Función analizadora de oraciones del backend.

```
"""# Analisis de emociones"""
# creación de la función analizadora de oraciones:
def find_sentences_with_noun_verb(request):
    text = request.POST.get('text')
    if text[-1] != '.':
        text += '.'
    translated = str(GoogleTranslator(source='auto', target='en').translate(str(text)))
    nlp = spacy.load("en_core_web_sm")
    doc = nlp(translated)
    for sent in doc.sents:
        if sent[0].is_title and sent[-1].is_punct:
            has_noun = 2
            has_verb = 1
            for token in sent:
                if token.pos_ in ["NOUN", "PROPN", "PRON"]:
                    has_noun -= 1
                elif token.pos_ == "VERB":
                    has_verb -= 1
            if has_noun < 1 and has_verb < 1:
                return JsonResponse({'message': 'True'})
    return JsonResponse({'message': 'False'})
```

Fuente: elaboración propia.

3. **Lectura y filtrado de datos:** para ofrecer recomendaciones de películas personalizadas, AngellA se basa en el conjunto de datos de películas estructurado previamente. La función *read_dataframe()* (ver Figura 40) se encarga de cargar este archivo de datos en forma de un *dataframe*. Este contiene información sobre películas, incluyendo detalles como género, sinopsis, emociones asociadas, entre otros.

Además, el sistema tiene en cuenta la edad del usuario para proporcionar recomendaciones apropiadas. La función *filtrar_edad()* realiza esta tarea (ver Figura 41), al filtrar las películas en el *dataframe* según la clasificación de edad y la adecuación de contenido para esa clasificación. Esto garantiza que las recomendaciones sean adecuadas y seguras para el usuario.

Figura 40. Función para cargar el conjunto de datos de películas del backend.

```
"""# Leer dataset de películas"""
# creación de la función para leer dataset:
def read_dataframe():
    df = pd.read_csv('core/chatbot/dataframe/df_movies_emotions_sparrows_cluster.csv', sep='^')
    peliculas = {}
    for index, row in df.iterrows():
        year = int(row['year']) # Convertir el año a entero antes de concatenar
        nombre_pelicula = row['name_latino']
        clasificacion = row['movie Rated']
        cluster = row['cluster_10']
        valores_emociones = row[['sadness', 'joy', 'love', 'anger', 'fear', 'surprise']].values
        peliculas[nombre_pelicula, str(year), clasificacion, cluster] = valores_emociones
    return peliculas
```

Fuente: elaboración propia.

Figura 41. Función para filtrar el conjunto de datos por edades del backend.

```
"""# Filtrar dataset por edades"""
# creación de la función para filtrar dataset por edades:
def filtrar_edad(edad, peliculas):
    clasificacion = {
        0: ['G', 'Approved', 'Passed', 'Not Rated'],
        1: ['PG', 'TV-PG', 'M/PG'],
        2: ['PG-13', 'TV-14'],
        3: ['R'],
        4: ['NC-17', 'TV-MA']
    }

    clasif = -1
    if edad > 0 and edad < 8:
        clasif = 0
    elif edad == 8:
        clasif = 1
    elif edad > 8 and edad <= 16:
        clasif = 2
    elif edad == 17:
        clasif = 3
    else:
        clasif = 4

    lista_clasificacion = []
    for i in range(clasif + 1):
        lista_clasificacion.append(clasificacion[i])

    peli = {}

    valores_clasificacion = [valor for sublista in lista_clasificacion for valor in sublista]
    for clave in peliculas:
        tercer_valor_clave = clave[2]
        if tercer_valor_clave in valores_clasificacion:
            peli[clave] = peliculas[clave]
    return peli
```

Fuente: elaboración propia.

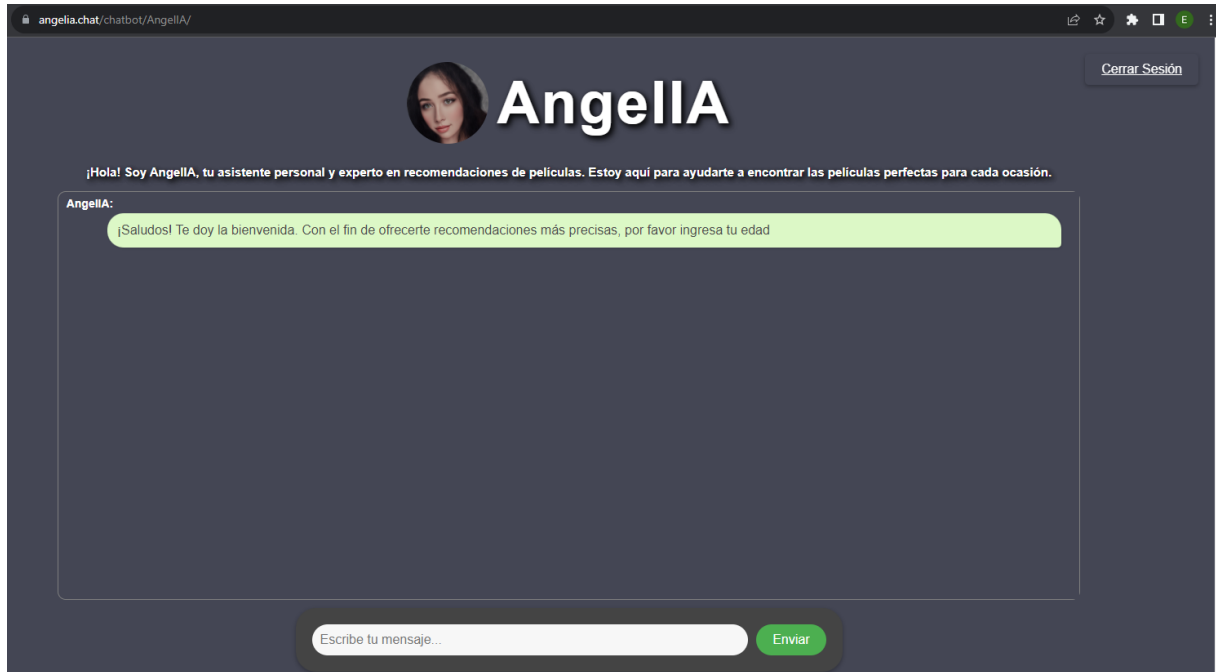
5.3.5. Frontend

La página *web* tiene un diseño minimalista con un encabezado que muestra el título y el avatar del *chatbot*, así como un enlace para cerrar sesión, como se muestra en la Figura 42.

El cuerpo de la página muestra el *chat* entre el usuario y el *chatbot*, presentado en forma de mensajes.

Un campo de entrada de texto permite al usuario escribir sus respuestas y un botón "Enviar" permite enviarlas al *chatbot*.

Figura 42. Página web del chatbot AngellA.



Fuente: elaboración propia.

1. **Desarrollo de la conversación:** AngellA inicia preguntando la edad del usuario. Esta debe ser en formato numérico entero, en un rango de 10 a 100 años, de lo contrario, AngellA solicita volver a digitar la edad. Este dato es fundamental para realizar el filtrado del conjunto de datos por edades para adecuar las recomendaciones.

Luego, dependiendo de la respuesta del usuario, hace preguntas sobre cómo ha sido su día, su estado de ánimo, y eventos recientes en su vida. Estas desempeñan un papel crucial en la identificación y extracción de las emociones y sentimientos de los usuarios.

De acuerdo con lo anterior, la formulación y justificación de las preguntas que hace AngellA a los usuarios son las siguientes:

1.1. ***“Cuéntame de forma resumida, ¿Cómo ha transcurrido tu día hasta ahora y qué has hecho?”***

Esta pregunta busca obtener una visión general del estado de ánimo y las actividades del usuario en el transcurso del día. Las respuestas proporcionan información sobre los eventos que pueden haber influido en el estado emocional del usuario. Además, la

temporalidad del "hasta ahora" sugiere una inmediatez que puede capturar emociones frescas y actuales, lo que resulta esencial para el análisis emocional en tiempo real.

1.2. *"Si tu ánimo fuera una pequeña historia de dos oraciones, ¿cómo la describirías?"*

Al solicitar al usuario que resuma su estado de ánimo en forma de historia, se le invita a reflexionar y condensar sus sentimientos actuales de una manera narrativa. Esto no sólo facilita la extracción de emociones específicas, sino que también puede revelar la intensidad y profundidad de dichas emociones. El formato breve de "dos oraciones" garantiza respuestas concisas, permitiendo al sistema procesar la información de manera eficiente.

1.3. *"Redacta una pequeña historia que te haya pasado últimamente."*

Con esta pregunta, se busca que el usuario comparta experiencias recientes que puedan estar influenciando su estado emocional. Estas narrativas pueden revelar eventos significativos, emociones persistentes o preocupaciones actuales. A través de esta pregunta, se obtiene una visión más detallada y contextual de la situación emocional del usuario.

AngellA procesa las respuestas del usuario y avanza a la siguiente pregunta en la secuencia. Una vez obtenidas todas las respuestas del usuario se almacenan en un arreglo en el *backend* para ser procesado para el análisis de emociones y generar las recomendaciones de películas. Finalmente, se pregunta al usuario si quiere recibir más recomendaciones. Si el usuario responde "Si", se generan nuevas recomendaciones y si responde "No" se termina la conversación y desconecta el flujo de trabajo. Para volver a conectar el flujo, debe darse clic al botón conectar. El flujo conversacional se puede ver en la Figura 43.

- 2. Recomendaciones de películas:** cuando AngellA recopila la información de los usuarios, genera recomendaciones de películas basadas en esas respuestas, como se muestra en la Figura 44.

Figura 43. Flujo conversacional del chatbot AngellA.



Fuente: elaboración propia.

Figura 44. Recomendaciones de películas del chatbot AngellA.



Fuente: elaboración propia.

Durante la generación de recomendaciones, se muestra una animación de carga. Las recomendaciones se presentan con detalles como el nombre de la película, su año de lanzamiento y un enlace para obtener más información.

El usuario puede calificar las recomendaciones dando "Me gusta" o "No me gusta". Estas calificaciones se guardan en una base de datos *Postgres* para posterior análisis.

5.3.6. Registro del usuario en el chatbot AngellA

Previamente al uso de AngellA, el usuario se debe registrar para poder acceder a este, como se muestra en la Figura 45. El registro tiene los siguientes campos obligatorios:

1. **Nombre de usuario:** es el nombre para iniciar sesión en el aplicativo.
2. **Nombre:** Nombre del usuario.
3. **Apellido:** Apellido del usuario.
4. **Email:** correo electrónico del usuario.
5. **Contraseña:** una contraseña segura formada por caracteres alfanuméricos también debe contener al menos un carácter especial como lo es punto, coma o asterisco y una letra mayúscula.

Figura 45. Página de registro de usuario en el chatbot AngellA.

AngellA

Registro de usuario

Nombre de usuario:

Nombre:

Apellido:

Email:

Contraseña:

Confirmación de contraseña:

☐ [Estoy de acuerdo con los términos y condiciones y política de tratamiento de datos](#)

Registrar

¿Ya tienes una cuenta? [Iniciar Sesión](#)

Fuente: elaboración propia.

Para completar el registro el usuario debe aceptar los términos y condiciones y política de tratamiento de datos, como se muestra en la Figura 46. Una vez completado el registro, se redirige a la vista principal de AngellA.

Para efectos de pruebas se ha creado el usuario **unir** con contraseña **Tfmunir2023*** con el cual puede accederse libremente a la página web para verificación de la funcionabilidad del sistema.

Figura 46. Términos y condiciones de uso del chatbot AngellA.

Términos y Condiciones de Uso

Al registrarte en AngellA, aceptas los siguientes términos y condiciones:

- Información del Usuario: Te comprometes a proporcionar información real, actualizada y completa como se solicita en el formulario de registro.
- Uso de la Plataforma: Aceptas utilizar AngellA de una manera ética, legal y no perjudicial para otros usuarios o la plataforma misma.
- Seguridad: Es tu responsabilidad mantener la confidencialidad de tu contraseña y cualquier actividad que ocurra en tu cuenta.

Política de Tratamiento de Datos

Al proporcionar tus datos en AngellA, entiendes y aceptas lo siguiente:

- Recolección de Datos: Recolectamos tu nombre de usuario, nombre, apellido, email, contraseña (encriptada) y edad para proporcionar y mejorar nuestros servicios.
- Uso de Datos: Tu información solo se usará para propósitos internos, como autenticación, comunicaciones y mejora del servicio. No venderemos ni compartiremos tus datos con terceros sin tu consentimiento explícito.
- Seguridad: Tomamos medidas para proteger tus datos, pero no podemos garantizar su seguridad absoluta. Si detectamos alguna brecha, te informaremos lo antes posible.
- Derechos: Tienes derecho a acceder, modificar o eliminar tu información personal en cualquier momento contactando a nuestro equipo de soporte.

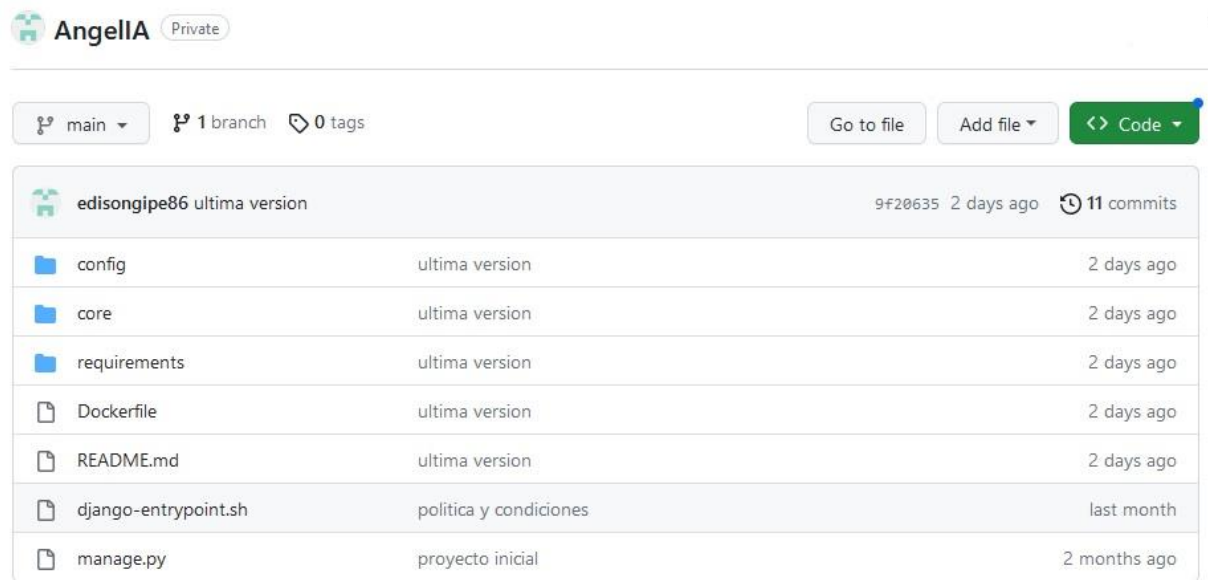
Aceptar

Fuente: elaboración propia.

5.3.7. Control de versiones del software

En el desarrollo de AngellA, se implementó un sistema de control de versiones utilizando el repositorio GitHub. Este enfoque no solo permitió rastrear y gestionar cambios en el código fuente a lo largo del tiempo, sino que también facilitó la colaboración entre los desarrolladores involucrados en el proyecto. La utilización de GitHub aseguró una administración sistemática y coherente de las modificaciones, garantizando la integridad y estabilidad del *software* a lo largo de sus distintas fases de desarrollo (ver Figura 47).

Figura 47. Repositorio GitHub del chatbot AngellA.



Fuente: elaboración propia.

5.4. Despliegue del sistema de chatbot: AngellA

5.4.1. Configuración del servidor

1. **Creación de una Instancia en Google Cloud Platform:** la configuración de un servidor en la Plataforma de GCP requiere el establecimiento de una instancia virtual, denominada igualmente como máquina virtual, en la infraestructura de dicha plataforma. Debido a consideraciones de seguridad (direcciones IP y servidores proxy, entre otros), se limitará la exposición al procedimiento de creación de esta instancia. Para ello, fue necesario ingresar a la Consola de GCP, seleccionar el proyecto donde instaura la instancia o, en su defecto, crear un proyecto nuevo si no existe uno. Posteriormente, se accedió a "*Compute Engine*" a través del menú de navegación, y finalmente, se seleccionó "Instancias de VM" y se presionó el botón "Crear instancia" para iniciar con la configuración adecuada.
2. **Configuración de la instancia:** se ingresó un nombre para la instancia. A continuación, se seleccionó una región o zona en la cual se ubica dicha instancia. Se optó por una plantilla de instancia predefinida basada en necesidades específicas, tales como sistema operativo, tipo de máquina y recursos. Se determinó la cantidad de CPU y memoria en función de los requisitos de la aplicación correspondiente. Finalmente, se configuraron las opciones de almacenamiento, contemplando tanto el disco de arranque como los discos adicionales, cuando fue pertinente. (ver Tabla 7).

Tabla 7. Especificaciones de la máquina virtual de GCP.

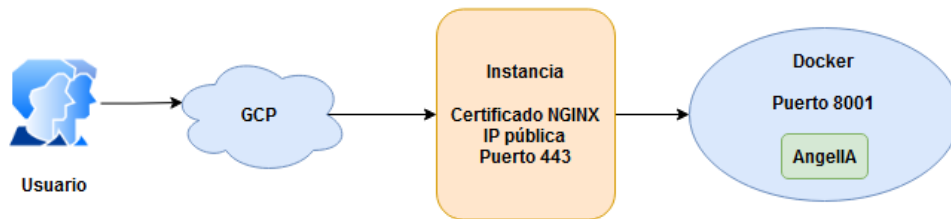
Necesidades específicas	Valor
Sistema Operativo	Ubuntu 22.04 LTS
Tipo de máquina virtual	Custom-1-7680-ext arq. x86-64
CPU	Intel Haswell
RAM	7.3 GB
Almacenamiento	9 GB

Fuente: elaboración propia.

3. **Acceso y seguridad, opciones de acceso Secure Shell:** se agregaron claves SSH (*Secure Shell*, por sus siglas en inglés) públicas para garantizar un acceso seguro a la instancia. De manera opcional, se habilitó el acceso de cuentas de servicio, previendo la utilización de las API de GCP desde la mencionada instancia.
4. **Configurar las reglas del *firewall*:** se definieron las reglas de *firewall* para permitir o denegar el tráfico de red hacia y desde la instancia. Por ejemplo, abrir puertos específicos como HTTP (80) o HTTPS (443).
5. **Revisión y creación:** se revisó la configuración de la instancia en la página de resumen. Tras constatar que todo estaba conforme, se procedió a pulsar "Crear" para iniciar la creación de la instancia.
6. **Acceso a la instancia:** una vez que se finalizó la creación de la instancia, fue posible acceder a ella a través de SSH desde la consola de GCP o mediante una herramienta de cliente SSH.
7. **Gestión y monitoreo:** se utilizaron las herramientas de monitoreo de GCP para supervisar el rendimiento y el uso de recursos de las instancias. Asimismo, se crearon instantáneas de disco con el objetivo de respaldar las instancias y tener la capacidad de restaurarlas ante eventuales inconvenientes.

Todas las configuraciones anteriores de la arquitectura de red se pueden ver en la Figura 48.

Figura 48. Arquitectura de red del servidor para el chatbot *AngellA*.



Fuente: elaboración propia.

5.4.2. Configuración de dominio

El dominio fue registrado bajo el nombre "*angelia.chat*" a través del proveedor *GoDaddy*. El proceso de configuración se desarrolló en las siguientes etapas:

1. En primer lugar, se inició sesión en *GoDaddy*. A continuación, se accedió a la configuración de DNS navegando hacia la sección de "Dominios" y seleccionando el dominio "*angelia.chat*" de entre los dominios registrados. En la página de detalles de dicho dominio, se localizó y accedió a la opción "Administrar DNS", que se encuentra bajo las etiquetas como "Configuración de DNS". Una vez en la configuración de DNS, se buscó la funcionalidad para agregar o modificar registros.
2. Para que "*angelia.chat*" apuntase a la dirección IP de la instancia en GCP, se añadió un registro tipo A, dejando el nombre en blanco e ingresando la dirección IP estática de la instancia en GCP. Tras ingresar los valores adecuados, se procedió a guardar los cambios. Se tuvo en cuenta que la propagación DNS podría requerir cierto tiempo, variando entre minutos y horas. Finalmente, una vez completada la propagación DNS, se realizó una verificación accediendo al dominio "angelia.chat" o "<https://angelia.chat>" a través de un navegador web.

5.4.3. Creación de Dockerfile

Un archivo *Dockerfile* (ver Figura 49) se crea para definir y describir cómo se debe construir una imagen de *Docker* para una aplicación específica. Una imagen de *Docker* es una plantilla que contiene todo lo necesario para ejecutar una aplicación, incluyendo el sistema operativo, las librerías, las dependencias y el código de la aplicación en sí. El *Dockerfile* proporciona instrucciones paso a paso sobre cómo configurar y ensamblar esa imagen.

Figura 49. Archivo *Dockerfile* que contiene las instrucciones de configuración para despliegue en el servidor.

```
ARG ImageID=python:3.11
FROM $ImageID

RUN apt-get update

ENV PYTHONUNBUFFERED 1

WORKDIR /app

COPY django-entrpoint.sh /django-entrpoint.sh
COPY . /app

RUN chmod +x /django-entrpoint.sh

RUN pip install -r requirements/requirements.txt

EXPOSE 8001

CMD ["/django-entrpoint.sh"]
```

Fuente: elaboración propia.

Las motivaciones para la creación de un archivo *Dockerfile* se sustentaron en las siguientes premisas:

1. **Reproducibilidad:** el archivo *Dockerfile* detalló meticulosamente el proceso de construcción de una imagen de *Docker*, asegurando que cualquier usuario con acceso al mismo pudiera replicar la imagen en distintos contextos. Esto consolidó la coherencia en la reproducción de entornos de desarrollo, pruebas y producción.
2. **Versión y control de cambios:** se almacenó el archivo *Dockerfile* en un sistema de control de versiones, como *GitHub*, junto al código fuente de la aplicación. Esto respaldó un control riguroso sobre las modificaciones en el ámbito de construcción, promoviendo la sinergia entre desarrolladores.
3. **Automatización:** la construcción automatizada de imágenes a través del archivo *Dockerfile* se integró en procedimientos de Integración Continua/Entrega Continua (CI/CD, por sus siglas en inglés), asegurando una construcción y prueba automáticas de cada modificación en un ambiente regulado antes de su implementación.

4. **Gestión de dependencias:** el archivo *Dockerfile* especificó las dependencias y ajustes del sistema esenciales para el funcionamiento de una aplicación, evitando conflictos de incompatibilidad y certificando la uniformidad de dependencias en todos los entornos.
5. **Portabilidad:** al tratarse de imágenes *Docker*, éstas mostraron portabilidad, permitiendo su ejecución en cualquier sistema con *Docker* instalado. El archivo *Dockerfile* garantiza que otros desarrolladores o equipos puedan operar la aplicación sin contratiempos relacionados con ajustes ambientales.
6. **Documentación:** se priorizó un archivo *Dockerfile* exhaustivamente comentado, funcionando como una documentación incorporada sobre el proceso de configuración y construcción de la imagen, lo que resultó esencial para entender decisiones configurativas y dependencias requeridas.

5.4.4. Construcción y ejecución de la imagen en Docker

Una vez que esté listo el archivo *Dockerfile*, se puede construir la imagen con el comando en la terminal de la Figura 50.

Figura 50. Construcción de la imagen en Docker para el chatbot AngellA.

```
root@zealous-almeida:/var/lib/docker/angellia# docker build . -t angellia
DEPRECATED: The legacy builder is deprecated and will be removed in a future release.
Install the buildx component to build images with BuildKit:
https://docs.docker.com/go/buildx/

Sending build context to Docker daemon  4.731MB
Step 1/12 : ARG ImageID=python:3.11
Step 2/12 : FROM $ImageID
--> a5fee9aa0e3a
Step 3/12 : RUN apt-get update
--> Using cache
--> 743c19e43ba5
Step 4/12 : ENV PYTHONUNBUFFERED 1
--> Using cache
--> 3817e6862f1b
Step 5/12 : WORKDIR /app
--> Using cache
--> 2c02dd2b0c10
Step 6/12 : COPY django-entrpoint.sh /django-entrpoint.sh
--> Using cache
--> 0040dbcc1552
Step 7/12 : COPY . /app
--> 8eb29b2910c8
Step 8/12 : RUN chmod +x /django-entrpoint.sh
--> Running in 1cf13ac30d34
Removing intermediate container 1cf13ac30d34
--> 72a735cb6584
Step 9/12 : RUN apt-get update && apt-get install -y libenchant-2-dev
--> Running in 29e442596c52
```

Fuente: elaboración propia.

El parámetro “-t” permite asignar un nombre y una etiqueta (*tag*) a la imagen. El “.” al final del comando indica que el contexto para la construcción de la imagen es el directorio actual.

Finalmente, se ejecutó el contenedor a partir de la imagen creada, en este caso “angellia” con ciertas opciones de configuración, como se ve en la Figura 51.

Figura 51. Ejecución de la imagen en Docker para el chatbot *Angelia*.

```
root@zealous-almeida:/var/lib/docker/angelia# docker run -d -p 8001:8001 --restart=always angelia
1e015a2f38ef8b60277b4ba6032d04e35cd8c124f061fd82531671c4dc48705a
root@zealous-almeida:/var/lib/docker/angelia#
```

Fuente: elaboración propia.

Las configuraciones adoptadas fueron las siguientes:

1. **-d:** se optó por la opción "*detached*" (desacoplado). Al emplear **-d**, el contenedor se ejecutó en segundo plano, liberando la terminal para otras operaciones en vez de quedar restringida por la salida del contenedor.
2. **-p 8001:8001:** esta configuración permitió mapear el puerto 8001 del sistema anfitrión al puerto 8001 del contenedor. De esta forma, cualquier tráfico dirigido al puerto 8001 del anfitrión fue redirigido al puerto 8001 del contenedor, esencial cuando una aplicación interna al contenedor opera escuchando un puerto determinado y se busca acceder a la misma desde un equipo externo.
3. **--restart=always:** se determinó que *Docker* debía reiniciar el contenedor de manera automática si éste cesaba su operación por algún motivo. El parámetro "*always*" garantiza que el contenedor se reanude continuamente, a no ser que se suspenda de manera manual.
4. **angelia:** corresponde al nombre de la imagen utilizada para generar el contenedor. *Docker* procede a buscar dicha imagen localmente y, en ausencia de esta, procede a descargarla de un registro público, como *Docker Hub*.

6. Evaluación del chatbot

6.1. Pruebas funcionales del sistema de recomendaciones

A continuación, se presentarán las pruebas realizadas mediante distintas situaciones de conversación con las que se busca enfatizar las recomendaciones proporcionadas por el sistema desde distintos contextos.

6.1.1. Evaluación de la influencia del sistema de traducción automática

El objetivo principal de esta situación es evaluar la variabilidad de las recomendaciones con y sin la traducción automática, con esto se pretende verificar que el modelo creado es capaz de proporcionar recomendaciones del mismo contexto y contenido emocional sin importar el

idioma de entrada. La Tabla 8 proporciona las preguntas y respuestas ingresadas al sistema mediante esta situación.

Tabla 8. Preguntas y respuestas en español e inglés ingresadas al chatbot.

Pregunta	Respuesta (original en español)	Respuesta (traducida al inglés mediante el traductor automático)
Cuéntame de forma resumida, ¿Cómo ha transcurrido tu día hasta ahora y que has hecho?	El día de hoy he estado trabajando y realmente he avanzado mucho en mi trabajo.	<i>Today I have been working and I have really made a lot of progress in my work.</i>
Si tu ánimo fuera una pequeña historia de dos oraciones, ¿cómo la describirías?	Con muchos nervios, pero feliz que las cosas van funcionando en la vida.	<i>With many nerves, but happy that things are working out in life.</i>
Redacta una pequeña historia que te haya pasado últimamente.	Hace poco fui al supermercado y encontré a un amigo que tenía muchos años de no ver. Fue super genial.	<i>I recently went to the supermarket and found a friend that I hadn't seen for many years. He was super cool.</i>

Fuente: elaboración propia.

El resultado de ingresar las respuestas anteriores se presenta en la Tabla 9. Se observa que las recomendaciones entre el modelo en español y en inglés presenta películas similares por lo que puede afirmarse que el modelo presenta adecuadamente respuestas en inglés y español.

Tabla 9. Recomendaciones realizadas a las respuestas en español e inglés ingresadas al chatbot.

Recomendaciones	Modelo en español	Modelo en inglés
Primeras recomendaciones del sistema	1. ¡Eso que tú haces! (1996)	1. ¡Eso que tú haces! (1996) -
	2. Uri: <i>The Surgical Strike</i> (2019)	2. Uri: <i>The Surgical Strike</i> (2019)
	3. El castillo en el cielo (1986)	3. El castillo en el cielo (1986)
	4. Los increíbles (2004)	4. Spider-Man: Un Nuevo Universo (2018)
	5. Spider-Man: Un Nuevo Universo (2018)	5. Cantando bajo la lluvia (1952)
Segundas recomendaciones del sistema	1. Cantando bajo la lluvia (1952)	1. Los increíbles (2004)
	2. Enredados (2010)	2. Enredados (2010)

3. Piratas del Caribe: La maldición de la Perla Negra (2003)	3. Los cazadores del arca perdida (1981)
4. Los cazadores del arca perdida (1981)	4. Piratas del Caribe: La maldición de la Perla Negra (2003)
5. El mago de Oz (1939)	5. El mago de Oz (1939)
Total de recomendaciones comunes entre ambos modelos	
10	

Fuente: elaboración propia.

6.1.2. Prueba del sistema en un contexto emocional altamente negativo

La Tabla 10 presenta una situación conversacional ficticia cargada de emociones altamente negativas con el propósito de analizar las recomendaciones presentadas.

Tabla 10. *Respuestas en español altamente negativas ingresadas al chatbot.*

Pregunta	Respuesta (original en español)
Cuéntame de forma resumida, ¿Cómo ha transcurrido tu día hasta ahora y que has hecho?	Me siento muy solo, siento que nadie de las personas me entiende.
Si tu ánimo fuera una pequeña historia de dos oraciones, ¿cómo la describirías?	Muy aburrido y sin ganas de vivir. Quiero morir.
Redacta una pequeña historia que te haya pasado últimamente.	Perdí mi trabajo, odio a mis padres y mi novia me abandonó. No sirvo para nada.

Fuente: elaboración propia.

Las recomendaciones entregadas por el sistema se presentan en la Figura 52, Tabla 11 y Tabla 12, Se han resaltado los aspectos más relevantes de la sinopsis para evidenciar la pertinencia de las recomendaciones en vista de las emociones negativas percibidas y las recomendaciones entregadas con propósito de mejorar el estado de ánimo del usuario.

Figura 52. Recomendaciones realizadas por AngellA a respuestas altamente negativas.

Izquierda: primeras cinco recomendaciones. Derecha: Segundas cinco recomendaciones.

Recuerda calificar la recomendación de la película.	Recuerda calificar la recomendación de la película.
1. Tres colores: Azul (1993) - Ver detalles	1. Perdidos en Tokio (2003) - Ver detalles
2. Lo imposible (2012) - Ver detalles	2. El náufrago (2000) - Ver detalles
3. Somos Marshall (2006) - Ver detalles	3. El Hombre Elefante (1980) - Ver detalles
4. Bajo La Misma Estrella (2014) - Ver detalles	4. Las viñas de la ira (1940) - Ver detalles
5. Crepúsculo: Luna nueva (2009) - Ver detalles	5. 2001: Odisea Espacial (1968) - Ver detalles

Fuente: elaboración propia.

Tabla 11. Primeras recomendaciones para respuestas en español altamente negativas.

Recomendación	Géneros de la recomendación	Extractos relevantes de la sinopsis (TMDB)
Tres colores: Azul (1993)	Drama	<i>“Al recuperarse de sus lesiones, decide comenzar una nueva vida, independiente, solitaria y anónima, alejada de los privilegios que antes disfrutaba...”</i> <i>“Olivier está enamorado de ella desde hace muchos años y acaba convenciéndola para que termine el «Concierto para Europa», una ambiciosa obra inacabada de Patrice...”</i>
Lo imposible (2012)	Aventura, Drama, Suspenso	<i>“Nada es más poderoso que el espíritu humano...”</i>
Somos Marshall (2006)	Drama, Historia	<i>“tuvo que afrontar el difícil desafío de construir de nuevo el equipo de la Universidad de Marshall...”</i>
Bajo La Misma Estrella (2014)	Romance, Drama	<i>“un viaje de reafirmación de vida...”</i>
Crepúsculo: Luna nueva (2009)	Aventura, Fantasía, Drama, Romance	<i>“Bella intentará superar el abandono de Edward...”</i>

Fuente: elaboración propia.

Tabla 12. *Segundas recomendaciones para respuestas en español altamente negativas.*

Recomendación	Géneros de la recomendación	Extractos relevantes de la sinopsis (TMDB)
Perdidos en Tokio (2003)	Drama, Romance	<i>"Poco a poco se hacen amigos y, a medida que exploran la ciudad juntos, empiezan a preguntarse si su amistad podría transformarse en algo más..."</i>
El náufrago (2000)	Aventura, Drama	<i>" tras cuatro años de subsistencia Chuck aprende todas las técnicas de supervivencia mientras sufre la tortura de la soledad. La solución: arriesgar la vida adentrándose mar adentro..."</i>
El Hombre Elefante (1980)	Drama, Historia	<i>"Un cirujano victoriano rescata a un hombre muy desfigurado y maltratado por su "dueño"..."</i> <i>"Detrás de su fachada monstruosa, se revela a una persona de gran inteligencia y sensibilidad..."</i>
Las viñas de la ira (1940)	Drama	<i>"Para escapar al hambre y a la pobreza, la familia no tiene más remedio que emprender un larguísimo viaje lleno de penalidades con la esperanza de encontrar una oportunidad en California, la tierra prometida..."</i>
2001: Odisea Espacial (1968)	Ciencia ficción, Misterio, Aventura	<i>"unos simios descubren un monolito que los lleva a un estadio de inteligencia superior..."</i>

Fuente: elaboración propia.

En base a las recomendaciones presentadas en las tablas, se evidencia que los géneros que predominan para contrarrestar emociones altamente negativas incluyen drama, romance, aventura, suspenso y ciencia ficción. Estos géneros, según los extractos relevantes de las sinopsis, están asociados con tramas que abordan desafíos personales, superación, reafirmación de vida, evolución emocional y descubrimientos que pueden resultar enriquecedores para el usuario. Es relevante destacar que las películas sugeridas no solo presentan conflictos, sino también resoluciones o reflexiones que pueden contribuir a la mejora del estado emocional del usuario, permitiéndole confrontar y procesar sus propias emociones a través de la narrativa cinematográfica.

6.1.3. Prueba del sistema en un contexto emocional altamente positivo

La Tabla 13 presenta una situación conversacional ficticia cargada de emociones altamente positivas con el propósito de analizar las recomendaciones presentadas y realizar una comparación con las situaciones negativas anteriormente presentadas.

Tabla 13. *Respuestas en español altamente positivas ingresadas al chatbot.*

Pregunta	Respuesta (original en español)
Cuéntame de forma resumida, ¿Cómo ha transcurrido tu día hasta ahora y que has hecho?	Me siento extremadamente feliz, todas las cosas estás marchando super bien en mi vida.
Si tu ánimo fuera una pequeña historia de dos oraciones, ¿cómo la describirías?	Feliz de superar cada uno de mis problemas y con mucho optimismo de mi futuro.
Redacta una pequeña historia que te haya pasado últimamente.	Últimamente he visitado a mis abuelos y estuve jugando videojuegos, fue una experiencia increíble.

Fuente: elaboración propia.

Figura 53. *Recomendaciones realizadas por AngellA a respuestas altamente positivas.*

Izquierda: primeras cinco recomendaciones. Derecha: Segundas cinco recomendaciones.

Recuerda calificar la recomendación de la película.	Recuerda calificar la recomendación de la película.
1. ¡Eso que tú haces! (1996) - Ver detalles	1. Cantando bajo la lluvia (1952) - Ver detalles
2. Los increíbles (2004) - Ver detalles	2. Enredados (2010) - Ver detalles
3. El castillo en el cielo (1986) - Ver detalles	3. Piratas del Caribe: La maldición de la Perla Negra (2003) - Ver detalles
4. Spider-Man: Un Nuevo Universo (2018) - Ver detalles	4. Los cazadores del arca perdida (1981) - Ver detalles
5. Uri: The Surgical Strike (2019) - Ver detalles	5. El mago de Oz (1939) - Ver detalles

Fuente: elaboración propia.

A partir de la información presentada en la Tabla 14 y Tabla 15, se evidencia que, en respuesta a emociones altamente positivas, el sistema de recomendación prioriza películas orientadas principalmente en los géneros de comedia, aventura, animación y acción. Estas selecciones se enfocan en tramas que realzan el éxito, la aventura y la superación. Sin embargo, es relevante destacar que el sistema también manifiesta cierto grado de aleatoriedad, debido a que los usuarios con emociones altamente positivas pueden estar propensos a explorar nuevas temáticas.

Tabla 14. *Primeras recomendaciones para respuestas en español altamente positivas.*

Recomendación	Géneros de la recomendación	Extractos relevantes de la sinopsis (TMDB)
¡Eso que tú haces! (1996)	Comedia, Drama, Romance, Música	<i>“Un grupo de rock desconocido se vuelve exitoso con la ayuda de un sensato empresario y una canción pegadiza...”</i>
Los increíbles (2004)	Acción, Aventura, Animación, Familia	<i>“Una familia de superhéroes encubiertos, mientras intentan vivir la tranquila vida suburbana, se ven obligados a entrar en acción para salvar el mundo...”</i>
El castillo en el cielo (1986)	Aventura, Fantasía, Animación, Acción, Familia, Romance	<i>“En esta aventura infantil, un joven minero y una misteriosa chica buscan una isla perdida que, según la leyenda, oculta grandes riquezas...”</i>
Spider-Man: Un Nuevo Universo (2018)	Acción, Aventura, Animación, Ciencia ficción	<i>“trae a una versión alternativa de Peter Parker que tratará de enseñarle a Miles como ser un mejor Spider-Man...”</i> <i>“4 versiones alternas de Spidey aparecerán y buscarán regresar a su universo antes de que toda la realidad colapse...”</i>
Uri: The Surgical Strike (2019)	Acción, Drama, Bélica	<i>“Una historia basada en un ataque militar de las Fuerzas Especiales Indias...”</i>

Fuente: elaboración propia.

Tabla 15. *Segundas recomendaciones para respuestas en español altamente positivas.*

Recomendación	Géneros de la recomendación	Extractos relevantes de la sinopsis (TMDB)
Cantando bajo la lluvia (1952)	Comedia, Romance	<i>“Antes de conocer a la aspirante a actriz Cathy Seldon, el ídolo del cine mudo Don Lockwood pensaba que lo tenía todo; fama, fortuna y éxito. Pero, tras conocerla, se da cuenta de que ella es lo que realmente faltaba en su vida...”</i>
Enredados (2010)	Animación, Familia	<i>“La mágica Rapunzel de pelo largo ha pasado toda su vida en una torre, pero ahora que un ladrón fugitivo se ha topado con ella, está a punto de descubrir el mundo por primera vez, y quién es realmente...”</i>
Piratas del Caribe: La maldición de la Perla Negra (2003)	Aventura, Fantasía, Acción	<i>“Will Turner, amigo de la infancia de Elizabeth, se une a Jack para rescatarla y recuperar el Perla Negra...”</i>

Los cazadores del arca perdida (1981)	Aventura, Acción	<i>“Indiana Jones es un profesor de arqueología, dispuesto a correr peligrosas aventuras con tal de conseguir valiosas reliquias históricas ...”</i>
El mago de Oz (1939)	Aventura, Fantasía, Familia	<i>“se embarca en una búsqueda con sus nuevos amigos para ver al Mago que puede ayudarla a volver a casa en Kansas y ayudar a sus amigos también...”</i>

Fuente: elaboración propia.

6.2. Pruebas realizadas por terceros

En el marco del presente estudio, se llevaron a cabo pruebas promocionando el sitio *web* *AngellA* a través de diversas plataformas de redes sociales. Durante un período comprendido entre el 20 y el 28 de agosto de 2023, se extendió una invitación a potenciales usuarios para que procedieran a registrarse y explorar el funcionamiento del sistema. Es importante mencionar que, dentro de la base de datos, únicamente quedaron registrados aquellos usuarios que efectuaron evaluaciones sobre las recomendaciones presentadas. Como resultado de esta fase, se contabilizaron 183 calificaciones aportadas por un conjunto de 68 usuarios diferentes.

Para el proceso de registro de las calificaciones aportadas por terceros, se empleó la interfaz proporcionada por *Django Admin*. Dicha interfaz, generada de forma automática, facilita a los desarrolladores la creación de una plataforma administrativa para sus aplicaciones *web*, eliminando la necesidad de diseñarla íntegramente desde su inicio.

Se creó un modelo entidad – relación con dos tablas de usuarios y calificaciones, como se ve en la Figura 54.

En la tabla Calificaciones se guarda la siguiente información:

1. Usuario.
2. Nombre de la película recomendada.
3. Año de la película.
4. Respuestas de las preguntas.
5. Edad del usuario.
6. Calificación de la película dada por el usuario.
7. Fecha y hora de creación del registro.

Figura 54. Modelo entidad - relación en el administrador de Django para las calificaciones de los usuarios.



Fuente: elaboración propia.

El acceso a estos registros está restringido a los administradores del sitio *web*. Por cuestiones de privacidad, una versión anónima de los registros sin el nombre de usuario se encuentra en *Hugging Face*².

Basado en los datos recopilados de las calificaciones de terceros se realizaron los siguientes análisis:

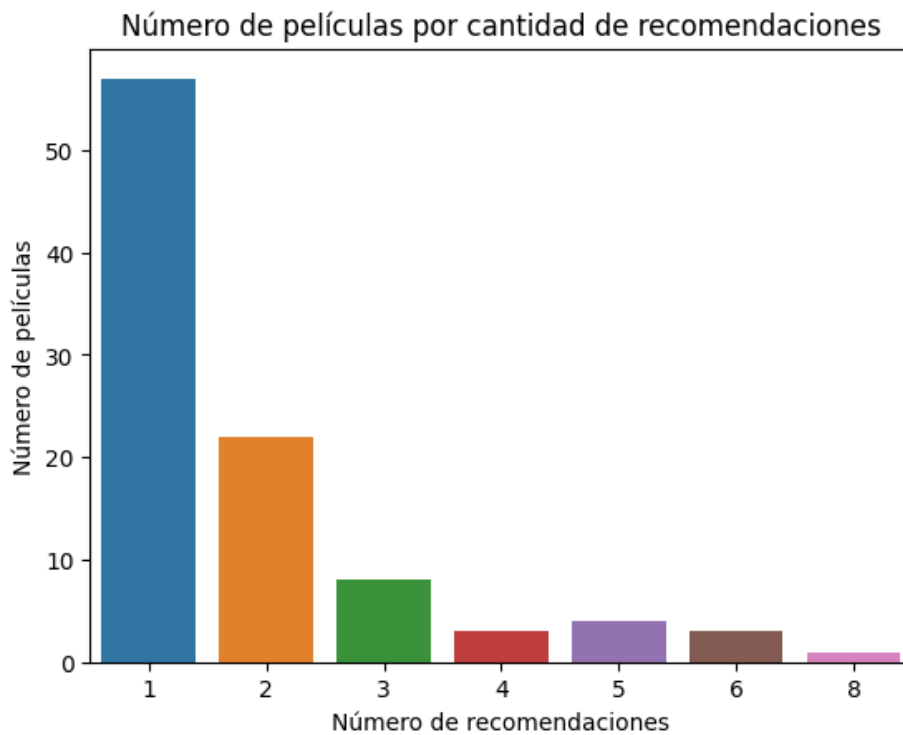
1. **Cantidad de recomendaciones por película:** esta evaluación posibilita la apreciación de la diversidad en las sugerencias proporcionadas por el *chatbot*. A través de este análisis, se busca determinar la repetitividad o unicidad de las películas recomendadas por el sistema. En la Figura 55, se observa que el *chatbot* ha realizado más de 50 recomendaciones diferenciadas, y en una proporción reducida ha ofrecido recomendaciones recurrentes. Esto demuestra que el sistema proporciona sugerencias diversificadas en función del perfil del usuario y sus interacciones correspondientes.
2. **Calificaciones de los usuarios:** tal como se detalló en la Sección 5, al usuario se le brinda la opción de evaluar las recomendaciones ofrecidas por el *chatbot*, seleccionando entre las opciones "Me gusta" y "No me gusta". Esta funcionalidad posibilita recopilar

² Registro de calificaciones de usuarios de *Hugging Face*: <https://huggingface.co/datasets/TFMUNIR/users-movies-ratings-28082023>

retroalimentación del usuario, la cual es esencial para la evaluación y optimización posterior del sistema.

En la Figura 56, se presenta un análisis detallado de las respuestas de los usuarios frente a las recomendaciones generadas por el *chatbot*. Se destaca que el 65% de las calificaciones, correspondiente a 118 respuestas, manifiestan conformidad con las películas sugeridas. Este resultado es significativo considerando que el sistema de recomendación del *chatbot* se fundamenta en el análisis emocional del usuario.

Figura 55. Número de películas por cantidad de recomendaciones.



Fuente: elaboración propia.

Figura 56. *Conteo de calificaciones de las películas proporcionadas por los usuarios.*



Fuente: elaboración propia.

Dicho enfoque, aunque innovador y centrado en una perspectiva más humana del usuario, puede no ser infalible. Esto se debe a la naturaleza subjetiva y fluctuante de las emociones humanas, lo que puede conducir a que, en determinadas ocasiones, algunas recomendaciones no sean del completo agrado del usuario. Es esencial reconocer esta característica al interpretar los resultados, entendiendo que el análisis emocional presenta tanto oportunidades como desafíos en el ámbito de las recomendaciones personalizadas.

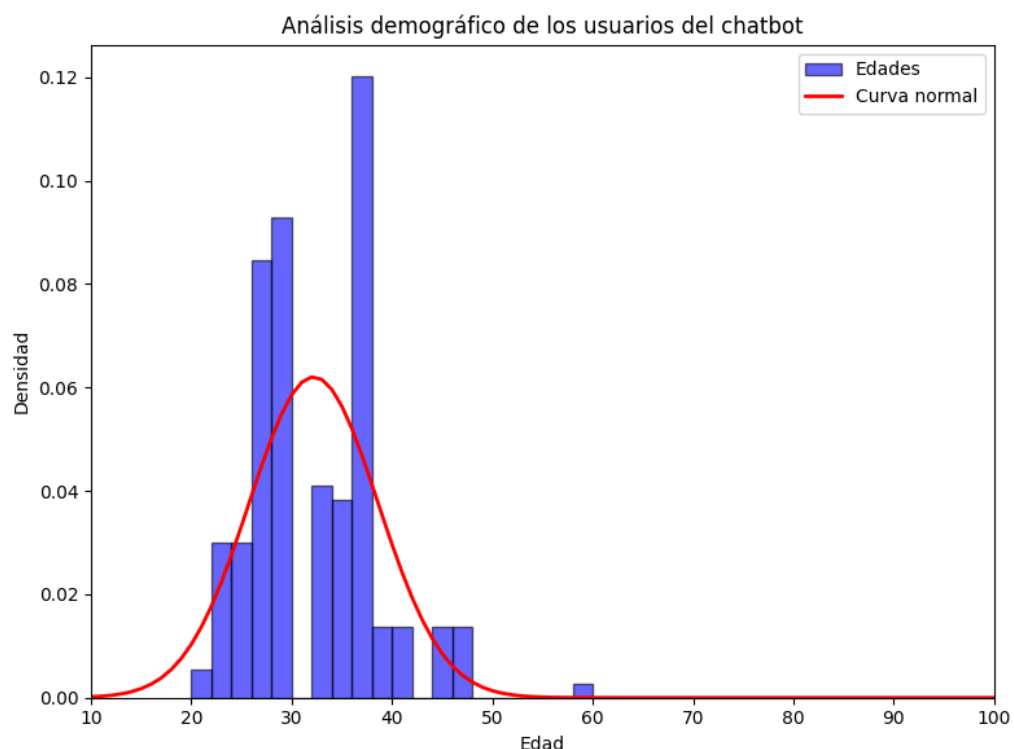
3. **Análisis demográfico de los usuarios:** se procedió a realizar un exhaustivo análisis de la edad de los usuarios, con el propósito esencial de determinar un perfil demográfico de los participantes que interactúan con el *chatbot*. Esta clasificación demográfica es fundamental, pues proporciona una clara comprensión de la diversidad de los usuarios. Con esta información, se busca entender y anticipar las potenciales preferencias y comportamientos de los usuarios en relación con las recomendaciones del *chatbot*. Cabe destacar que, si bien se ha identificado la importancia de la continua adaptación de la interfaz y optimizar las interacciones basadas en estos hallazgos, tales adaptaciones y mejoras se pueden considerar en investigaciones y desarrollos subsiguientes.

En la Figura 57, se presenta una distribución por edad de los usuarios que han establecido interacciones con el *chatbot*. En este gráfico se puede observar que un segmento

predominante de los usuarios se encuentra en el rango de edad comprendido entre los 20 y 40 años.

Este dato es indicativo de que el *chatbot* ha tenido una mayor resonancia o adaptabilidad dentro de este grupo demográfico, lo cual podría ser resultado de diversos factores, como la afinidad tecnológica, hábitos de consumo de películas o la propensión a utilizar sistemas automatizados de recomendación. Esta información es crucial para el entendimiento profundo del alcance y aceptación del *chatbot* para las diferentes edades, y sienta bases para futuras investigaciones o mejoras orientadas específicamente a usuarios fuera del rango de edades obtenido.

Figura 57. Análisis demográfico de los usuarios del chatbot.

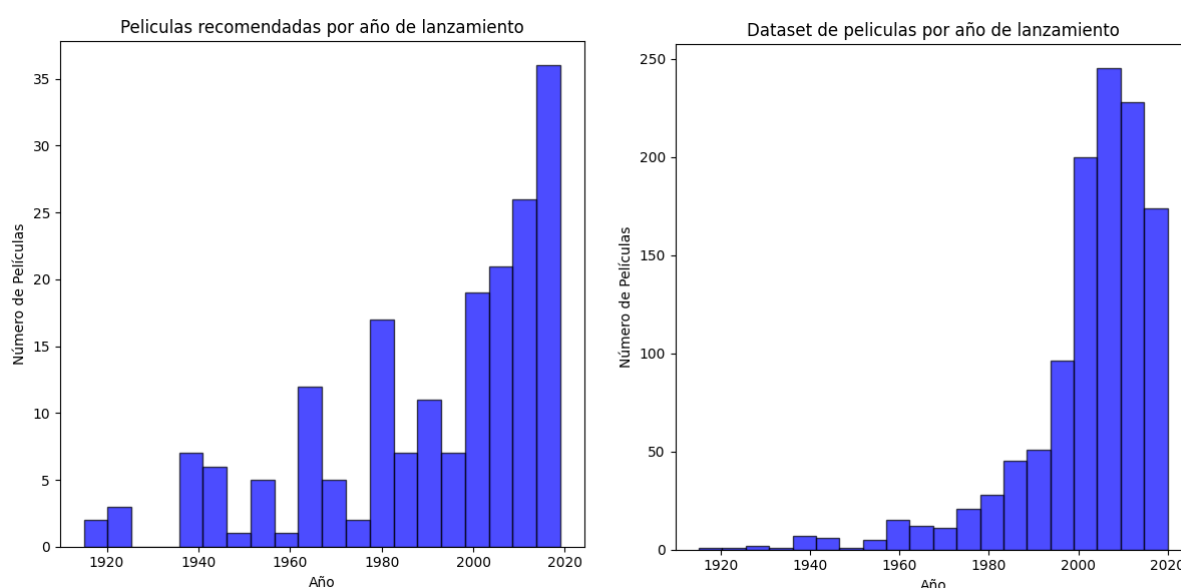


Fuente: elaboración propia.

- 4. Películas recomendadas por año de lanzamiento:** se aborda una evaluación de las sugerencias proporcionadas por el *chatbot* en función del año en que las películas fueron estrenadas. Esta perspectiva es importante para entender si las recomendaciones del sistema tienden hacia producciones más recientes o si, por el contrario, sugiere películas de épocas pasadas. Además, permite analizar si el *chatbot* mantiene un equilibrio en la oferta de títulos a lo largo de diferentes décadas o si presenta alguna inclinación particular.

En la Figura 58 presenta una comparativa entre las recomendaciones de películas ofrecidas por el *chatbot* y la distribución de películas contenida en el conjunto de datos de películas. De este análisis, se extrae que el comportamiento del *chatbot* sigue una tendencia alineada con la estructura del conjunto de datos. Es decir, existe una inclinación destacada hacia la recomendación de películas más reciente. Esta observación es indicativa de que el chatbot, tiene una predisposición a sugerir títulos más contemporáneos. Este hallazgo plantea oportunidades de mejora interesantes sobre la adaptabilidad del *chatbot* a diferentes usuarios, así como la importancia de mantener un equilibrio en las recomendaciones para garantizar una oferta diversificada.

Figura 58. Análisis de películas recomendadas por año de lanzamiento.



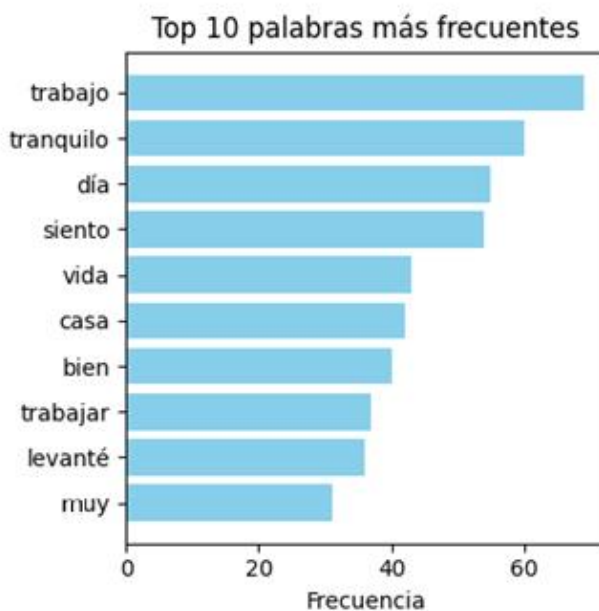
Fuente: elaboración propia.

Nube de palabras en las respuestas de los usuarios: la nube de palabras representa una herramienta gráfica que sintetiza y visualiza las palabras más recurrentes en las respuestas proporcionadas por los usuarios al interactuar con el *chatbot*. Esta representación permite identificar, de manera inmediata, los términos y conceptos que prevalecen en las interacciones, ofreciendo un panorama general de las temáticas o preocupaciones más comunes entre los participantes. La nube de palabras, al ponderar visualmente las palabras según su frecuencia, facilita la distinción de patrones y tendencias en el conjunto de datos, siendo de especial relevancia para comprender las dinámicas de comunicación

y las áreas de interés o inquietud de los usuarios en el contexto del sistema de recomendación implementado.

En la Figura 59, se presenta un análisis cuantitativo de las palabras más recurrentes en el conjunto de datos analizado. La palabra "trabajo" encabeza la lista con una frecuencia de 69, seguida de "tranquilo" con una frecuencia de 60 y "día" con una frecuencia de 55. Estas palabras son indicativas de las temáticas o contextos más mencionados por los usuarios. Otras palabras destacadas son "siento" (54 veces), "vida" (43 veces) y "casa" (42 veces). Se puede apreciar, igualmente, que términos como "bien" y "trabajar" han sido señalados 40 y 37 veces, respectivamente. Además, "levanté" y "Muy" también forman parte de este listado con frecuencias de 36 y 31.

Figura 59. Nube de palabras que aparecen con mayor frecuencia en el chatbot.

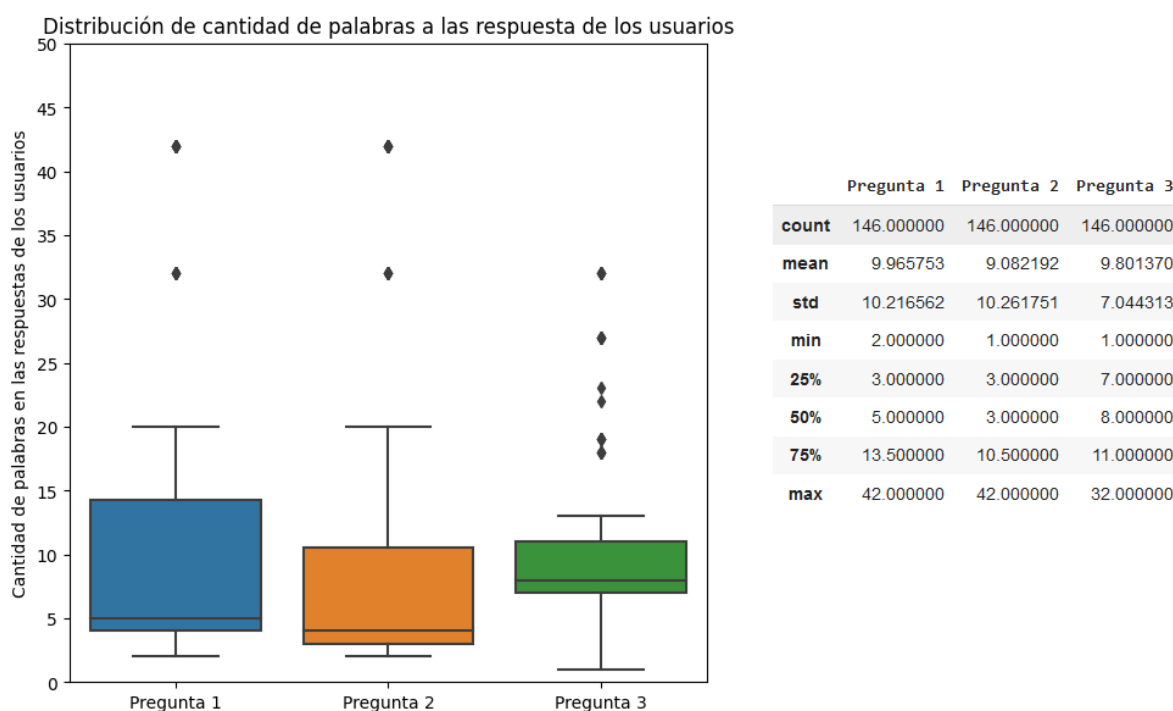


Fuente: elaboración propia.

5. **Cantidad de palabras en las respuestas de los usuarios:** en el proceso de análisis de las interacciones de los usuarios con el *chatbot*, es importante considerar no solo el contenido de las respuestas, sino también la extensión de estas.

Al medir el número de palabras que componen cada respuesta, se puede obtener una idea del nivel de detalle o profundidad con la que los usuarios están abordando las preguntas propuestas. Esta métrica puede reflejar, por ejemplo, el grado de compromiso del usuario, la claridad de la pregunta o incluso la capacidad del chatbot para generar recomendaciones más precisas. Una respuesta más extensa podría indicar una reflexión más profunda o una experiencia más rica por parte del usuario, mientras que una respuesta breve podría sugerir lo contrario.

Figura 60. Distribución de la cantidad de palabras en las respuestas de los usuarios.



Fuente: elaboración propia.

La Figura 60 muestra las estadísticas de la cantidad de palabras que tienen las respuestas de los usuarios a las preguntas del *chatbot*.

En el análisis de la distribución de palabras en las respuestas de los usuarios muestra que, en general, las respuestas tienden a ser breves, con una media alrededor de 10 palabras para las tres preguntas. Esto sugiere que los usuarios tienden a proporcionar respuestas concisas. Sin embargo, la notable variabilidad, especialmente en las preguntas 1 y 2, indica

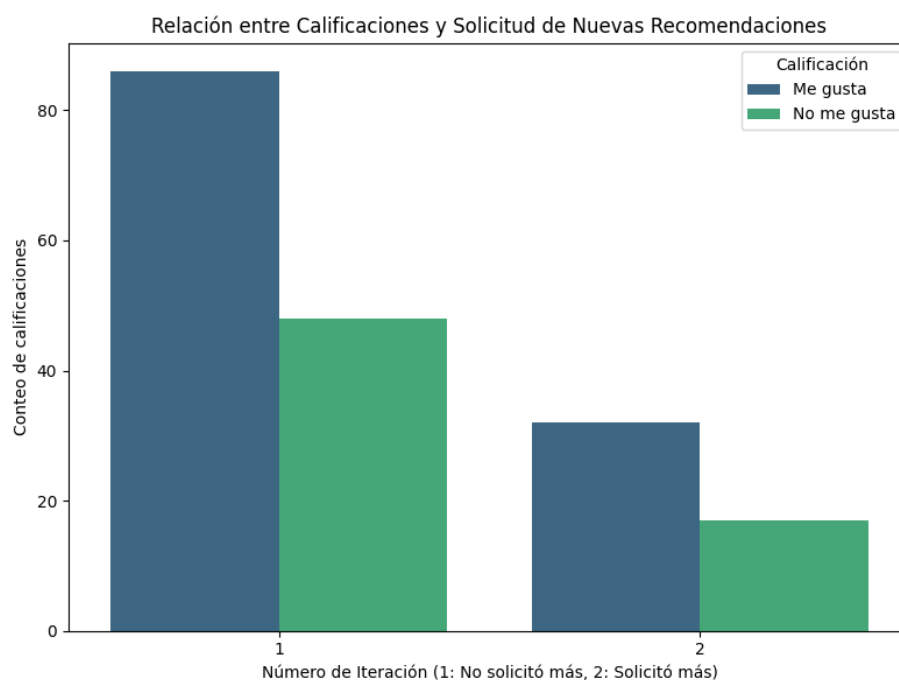
que hay un grupo de usuarios que proporciona respuestas mucho más detalladas, alcanzando hasta 42 palabras en algunos casos.

Es interesante observar que, aunque la pregunta 2 tiene un promedio ligeramente inferior en términos de longitud de respuesta en comparación con las preguntas 1 y 3, comparte el mismo valor máximo en la cantidad de palabras. Esto puede indicar que, si bien en general los usuarios podrían ser más concisos en la pregunta 2, todavía hay quienes se extienden en sus respuestas de manera similar a la pregunta 1.

La pregunta 3, aunque tiene un promedio similar a la pregunta 1, muestra una menor dispersión en las respuestas, lo que podría sugerir que los usuarios tienen una tendencia más uniforme en la longitud de sus respuestas para esta pregunta en particular.

6. **Relación de las calificaciones y solicitudes de nuevas recomendaciones por parte del usuario:** en el siguiente análisis, se explora la relación entre las calificaciones otorgadas por los usuarios a las películas recomendadas por el *chatbot* y la solicitud de nuevas recomendaciones por parte de los mismos usuarios. El propósito es determinar si existe una correlación entre la satisfacción del usuario, reflejada en sus calificaciones, y su deseo de obtener más recomendaciones de películas.

Figura 61. *Relación entre las calificaciones y la solicitud de nuevas recomendaciones hechas por los usuarios.*



Fuente: elaboración propia.

En la Figura 61 se puede observar que, de los usuarios que no solicitaron recomendaciones adicionales (primera iteración), 85 películas fueron calificadas positivamente con un "Me gusta", mientras que 45 fueron calificadas con un "No me gusta". Esto sugiere que, en general, las primeras recomendaciones ofrecidas por el chatbot fueron bien recibidas, con una tasa de aprobación que supera a las desaprobaciones en un ratio de 2:1.

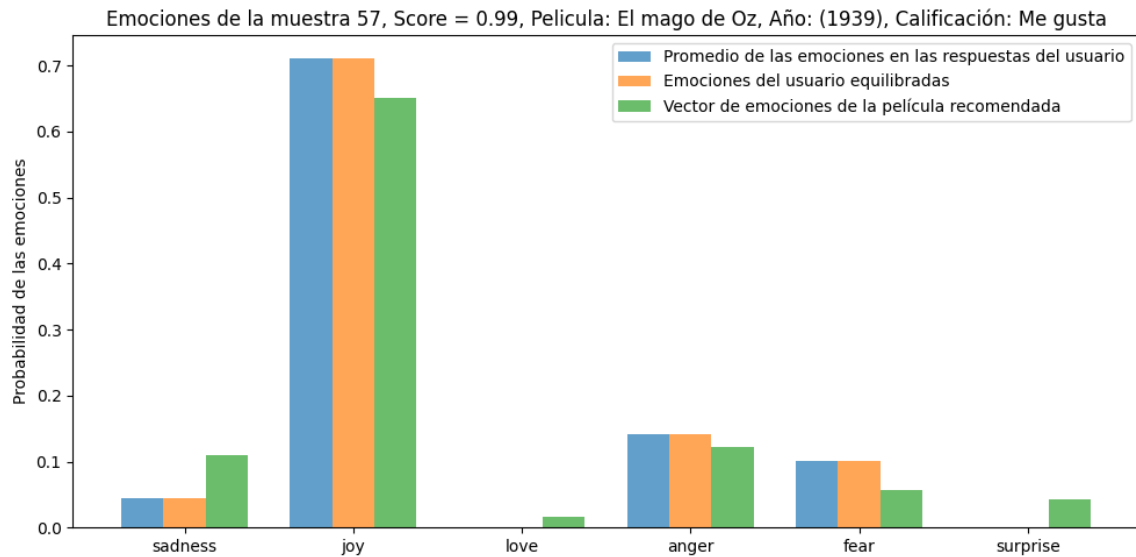
Por otro lado, en la segunda iteración, es decir, de los usuarios que expresaron el deseo de obtener más recomendaciones, 30 películas recibieron un "Me gusta" y 15 un "No me gusta". La proporción de calificaciones positivas respecto a las negativas sigue siendo de 2:1, similar a la observada en la primera iteración.

Estos resultados indican que la solicitud de recomendaciones adicionales no tuvo un impacto negativo significativo en la satisfacción del usuario con las películas recomendadas. Es más, el chatbot mantuvo una consistencia en la calidad de sus recomendaciones a lo largo de las iteraciones. Este hallazgo es crucial, ya que sugiere que el chatbot es capaz de mantener un nivel de precisión en sus recomendaciones incluso después de que se le solicite extender su lista de sugerencias.

7. **Comparativa visual de los vectores de emociones:** El siguiente análisis explica de forma visual el funcionamiento del sistema de recomendación en diferentes contextos emocionales para situaciones reales de uso. Como se mencionó en la Sección 5, las emociones del usuario luego de ser procesadas por el sistema son equilibradas por medio del factor de ajuste si se detecta alguna emoción predominantemente negativa por encima del umbral preestablecido.

La Figura 62 muestra que ante una emoción predominante positiva no realiza un reajuste del vector de emociones del usuario y pasa a la etapa de identificación de la película por medio de la distancia de coseno más cercana, que en este caso arroja un valor de similitud de 0.99. El usuario manifestó que la recomendación fue de su agrado.

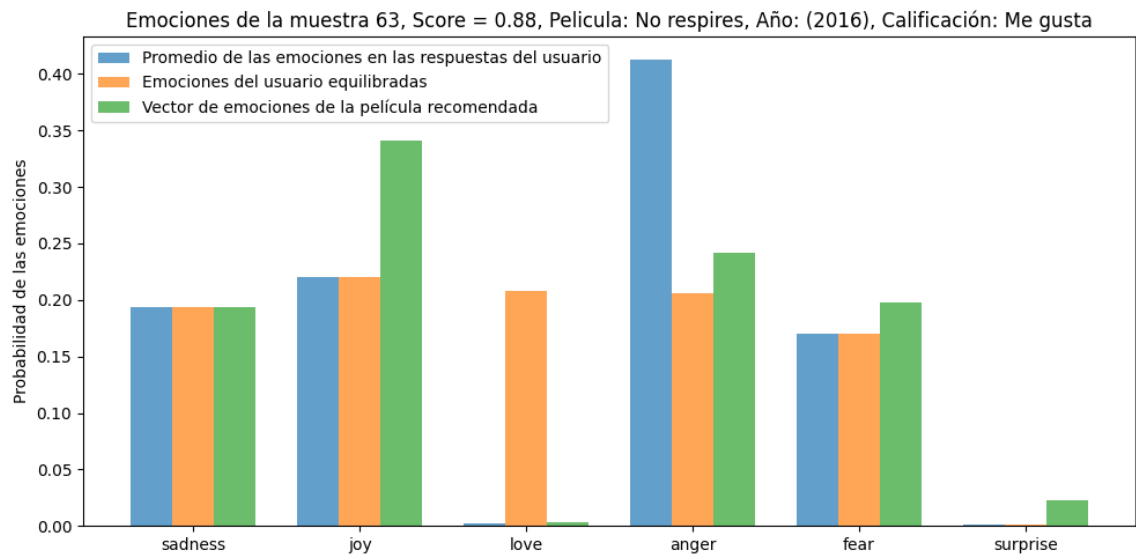
Figura 62. Recomendación de película con emoción del usuario altamente positiva.



Fuente: elaboración propia.

La Figura 63 ilustra una situación en la que el usuario experimenta una emoción combinada. El sistema modifica el vector de emociones del usuario, reduciendo la intensidad de las emociones negativas como tristeza, rabia y temor, y potenciando las positivas, específicamente alegría y amor. Luego, el sistema avanza a la fase de identificación de la película utilizando la distancia de coseno, donde se obtuvo una similitud de 0.88. De acuerdo con la retroalimentación del usuario, la sugerencia fue satisfactoria.

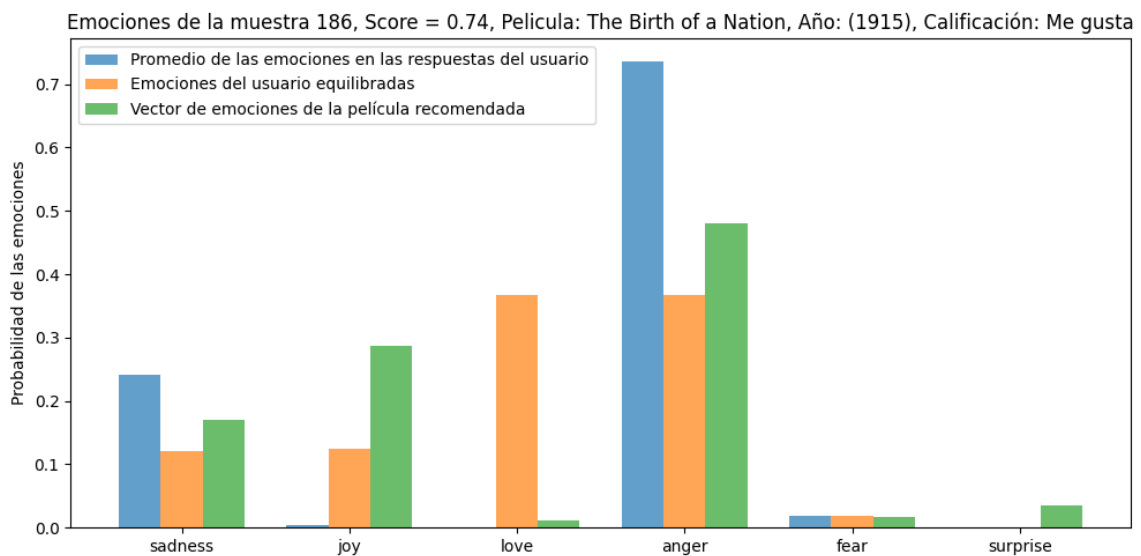
Figura 63. Recomendación de película con emociones mixtas del usuario.



Fuente: elaboración propia.

Según se puede observar en la Figura 64, ante la presencia de una emoción predominantemente negativa, el sistema recalibra el vector emocional del usuario, minimizando las emociones negativas (rabia y tristeza) y amplificando las positivas (alegría y amor). Una vez realizado este ajuste, la elección de la película se realiza a través del método de distancia de coseno, arrojando una similitud de 0.74 en este caso particular. La validación por parte del usuario indica que la elección propuesta fue coherente con sus expectativas.

Figura 64. Recomendación de película con emoción del usuario altamente negativa.



Fuente: elaboración propia.

7. Conclusiones y trabajos futuros

7.1. Conclusiones

Este proyecto de fin de máster destaca el poder del procesamiento del lenguaje natural en la mejora de la personalización de sistemas de recomendación mediante el análisis emocional. Se diseñó un sistema de recomendación de películas, fusionando estas características con las capacidades del chatbot AngellA.

Las contribuciones de este trabajo se reflejan en la creación de un conjunto de datos de películas específicamente orientado a las emociones detectadas por espectadores previos y a las características de cada una de ellas (año, género, clasificación y sinopsis), así como la implementación de un modelo de análisis emocional y la construcción del sistema de

recomendación integrado en el chatbot. Estos componentes se han diseñado para ofrecer recomendaciones de películas que se adapten mejor a las emociones de los usuarios, mejorando así la personalización y la satisfacción de estos.

La técnica de clusterización permitió agrupar películas con distintas variables sujetas a consideración, una de las cuales era el vector promedio de emociones de las reseñas realizadas por espectadores previos de las mismas, este enfoque innovador logró determinar un perfil emocional para cada película, lo que permitió al chatbot ofrecer recomendaciones a los usuarios basadas en la concordancia entre sus emociones y las evocadas por las películas.

La investigación también permitió encontrar emociones afines y opuestas entre sí, lo que llevó al desarrollo de un modelo de reajuste fácil de implementar, que no habría sido factible sin la detección adecuada de las emociones expresadas por los usuarios.

En términos de evaluación del modelo de detección de emociones, se entrenó exitosamente un modelo de transformers con alta precisión capaz de clasificar adecuadamente la mayoría de los estados de ánimo que fueron sujetos de estudio, como lo demostraron las métricas del modelo presentadas en la Sección 6, donde se obtuvieron valores de precisión para la detección de emociones por encima del 93%.

La evaluación realizada al sistema de recomendación ha demostrado pertinencia en la adaptación de las recomendaciones de películas a las emociones de los usuarios. Los resultados obtenidos en las pruebas funcionales internas y de usuario (terceros) respaldan la eficacia de la solución propuesta.

7.2. Trabajos Futuros

Finalizada esta investigación, se abren nuevas perspectivas para futuros desarrollos en el campo de la inteligencia artificial y el análisis emocional aplicado a sistemas de recomendación. Algunos de los caminos que podrían explorarse en futuras investigaciones incluyen:

1. **Interacción contextual avanzada:** en relación con la arquitectura del chatbot AngellA, es esencial seguir explorando formas de enriquecer su capacidad de interacción con los usuarios. Esto podría incluir el desarrollo de un sistema de diálogo más avanzado que sea capaz de comprender contextos complejos y mantener conversaciones más naturales. En

este sentido, las investigaciones pueden ir enfocadas a la generación de texto mediante los LLM actualmente en auge y cada vez más posibles de entrenarse con pocos recursos.

2. **Expansión del conjunto de datos de películas:** para incluir más variedad, novedad e información adicional que podría ser relevante para el entrenamiento de los modelos.
3. **Exploración de emociones:** se pueden explorar algoritmos de recomendación avanzados y estrategias de ajuste emocional que involucren más estados de ánimo y desarrollar modelos más avanzados que tengan en cuenta matices y contextos emocionales más complejos.
4. **Adaptación a otros dominios:** la adaptación de AngellA para aplicaciones en otros dominios, como la música, los libros, videojuegos o incluso la atención médica, podría ampliar su utilidad y potencial.
5. **Desarrollo de algoritmos de recomendación avanzados:** explorar algoritmos de recomendación más avanzados, como modelos de aprendizaje profundo y sistemas basados en conocimiento o aprendizaje por refuerzo orientado a proporcionar recomendaciones cada vez más precisas y orientadas a mejorar las emociones de los usuarios.
6. **Aplicaciones educativas:** investigar la aplicación de la adaptación emocional en sistemas de recomendación para la educación, ayudando a los estudiantes a encontrar recursos de aprendizaje que se adapten a su estado emocional y nivel de competencia.
7. **Evaluación de la experiencia del usuario:** realizar estudios más profundos sobre la experiencia del usuario con el sistema de recomendación y el chatbot para comprender mejor cómo afectan las emociones a la satisfacción del usuario y cómo se pueden realizar mejoras adicionales.
8. **Análisis emocional multimodal:** ampliar la capacidad de AngellA para realizar análisis de emociones de múltiples modalidades o fuentes de entrada. En lugar de confiar en una sola fuente de datos, como podría ser el texto, podría considerarse incluir audio, video, datos fisiológicos, entre otros, para identificar y analizar las emociones.

Este trabajo sentó las bases para una investigación y desarrollo continuos en la convergencia de la inteligencia artificial, el análisis emocional y los sistemas de recomendación. Las líneas de trabajo futuro aquí mencionadas representan oportunidades para seguir avanzando en este campo y aprovechar al máximo las contribuciones realizadas.

Referencias bibliográficas

- Adomavicius, G., & Tuzhilin, A. (2005). Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE Transactions on Knowledge and Data Engineering*, 17(6), 734-749.
- Ameta, U., Patel, M., & Rathore, N. S. (2023). Fusing Artificial Intelligence with Scrum Framework. *SSRN Electronic Journal*, 6.
- Arora, S., Narayan, A., Chen, M. F., Orr, L., Guha, N., Bhatia, K., Chami, I., Sala, F., & Ré, C. (2022). Ask Me Anything: A simple strategy for prompting language models. *ArXiv*, 63.
- Barrett, L. F. (2006). Solving the emotion paradox: categorization and the experience of emotion. *Personality and social psychology review : an official journal of the Society for Personality and Social Psychology, Inc*, 10(1), 20-46.
- Bennett, J., & Lanning, S. (2007). The Netflix Prize. *KDDCup'07*, 3-6.
- Bharti, S. K., Varadhaganapathy, S., Gupta, R. K., Shukla, P. K., Bouye, M., Hingaa, S. K., & Mahmoud, A. (2022). Text-Based Emotion Recognition Using Deep Learning Approach. *Computational Intelligence and Neuroscience*, 2022, 8.
- Binali, H., Wu, C., & Potdar, V. (2010). Computational approaches for emotion detection in text. *4th IEEE International Conference on Digital Ecosystems and Technologies (IEEE DEST 2010)*, 172-177.
- Blaus, B. (2014). Medical gallery of Blausen Medical 2014. *WikiJournal of Medicine*, 1(2), 78.
- Caldarini, G., Jaf, S., & McGarry, K. (2022). A Literature Survey of Recent Advances in Chatbots. *Information* 2022, 13(1), 41.
- Calvo, R., & D'Mello, S. (2010). Affect detection: An interdisciplinary review of models, methods, and their applications. *IEEE Transactions on Affective Computing*, 1(1), 18-37.
- Chapaval, N. (2017). Qué es Frontend y Backend: principal diferencia y ejemplos [Blog]. *Platzi*. <https://platzi.com/blog/que-es-frontend-y-backend/>

- Cremonesi, P., Garzotto, F., Negro, S., Papadopoulos, A. V., & Turrin, R. (2011). Comparative evaluation of recommender systems Quality. *Conference on Human Factors in Computing Systems*, 1927-1932.
- Cruse, D. A. (1986). *Lexical semantics*. Cambridge University Press.
- Dahiya, M. (2017). A Tool of Conversation: Chatbot. *International Journal of Computer Sciences and Engineering*, 5(5), 158-161.
- Das, A. (2021). Multi-label Emotion Classification with PyTorch + HuggingFace's Transformers and W&B for Tracking [Blog]. *Towards Data Science*. <https://towardsdatascience.com/multi-label-emotion-classification-with-pytorch-huggingfaces-transformers-and-w-b-for-tracking-a060d817923>
- Davidov, D., Tsur, O., & Rappoport, A. (2010). Enhanced Sentiment Learning Using Twitter Hashtags and Smileys. *Coling 2010 Organizing Committee*, 241-249.
- Davis, M., & Whalen, P. J. (2001). The amygdala: vigilance and emotion. *Molecular Psychiatry* 2001 6:1, 6(1), 13-34.
- Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *ArXiv*, 16.
- Django Software Foundation. (2023). Django Framework. *World Wide Web Consortium (W3C)*.
- ECMA International. (2023). ECMAScript® 2023 Language Specification. *ECMAScript® 2023 Language Specification*.
- Ekman, P. (1992). An argument for basic emotions. *Cognition and Emotion*, 6(3-4), 169-200.
- Ekman, P. (2005). Basic Emotions. *Handbook of Cognition and Emotion*, 45-60.
- Fadel, M. (2022). 5 Websites to Download Pre-trained Machine Learning Models [Blog]. *Towards Data Science*. <https://towardsdatascience.com/5-websites-to-download-pre-trained-machine-learning-models-6d136d58f4e7>
- Falk, K. (2019). Practical Recommender Systems. En *Manning*.

- Fernández-Paniagua, A. M. del C. (2018, julio 20). Filtrado colaborativo: ¿Qué es y para qué sirve? [Blog]. *IEBS*. <https://www.iebschool.com/blog/filtrado-colaborativo-sirve-e-commerce/>
- Gilbert, S., Harvey, H., Melvin, T., Vollebregt, E., & Wicks, P. (2023). Large language model AI chatbots require approval as medical devices. *Nature Medicine*, 1-3.
- Gnewuch, U., Morana, S., Adam, M., & Maedche, A. (2018). Faster is Not Always Better: Understanding the Effect of Dynamic Response Delays in Human-Chatbot Interaction. *Twenty-Sixth European Conference on Information Systems*, 19.
- Goodman, B., & Flaxman, S. (2016). European Union regulations on algorithmic decision-making and a «right to explanation». *AI Magazine*, 38(3), 50-57.
- Google. (2020). Security whitepapers. *Google Cloud*.
- Greyling, C. (2019). A First Look At Microsoft Power Virtual Agent [Blog]. *Medium*. <https://cobusgreyling.medium.com/a-first-look-at-microsoft-power-virtual-agent-4961311afe4a>
- GroupLens. (2023). *MovieLens*. GroupLens.
- Gunawan, D., Sembiring, C., & Budiman, M. (2018). The Implementation of Cosine Similarity to Calculate Text Relevance between Two Documents. *Journal of Physics: Conference Series*, 978, 012120.
- Guo, L., Liang, J., Zhu, Y., Luo, Y., Sun, L., & Zheng, X. (2019). Collaborative filtering recommendation based on trust and emotion. *Journal of Intelligent Information Systems*, 53(1), 113-135.
- Gupta, A. (2022). Best Chatbot Frameworks You Must Know About [Blog]. *Medium*. <https://medium.com/@letthedataconfess/best-chatbot-frameworks-you-must-know-about-6926d9e4d80f>
- Herlocker, J. L., Konstan, J. A., Terveen, L. G., & Riedl, J. T. (2004). Evaluating collaborative filtering recommender systems. *ACM Transactions on Information Systems (TOIS)*, 22(1), 5-53.
- Hosna, A., Merry, E., Gyalmo, J., Alom, Z., Aung, Z., & Azim, M. A. (2022). Transfer learning: a friendly introduction. *Journal of Big Data*, 9(1), 102.

- Howard, J., & Ruder, S. (2018). Universal Language Model Fine-tuning for Text Classification. *Fifty-sixth Annual Meeting of the Association for Computational Linguistics*, 328-339.
- Hu, M., & Liu, B. (2004). Mining and summarizing customer reviews. *Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 168-177.
- Ibrahim, M., Bajwa, I. S., Ul-Amin, R., & Kasi, B. (2019). A Neural Network-Inspired Approach for Improved and True Movie Recommendations. *Computational Intelligence and Neuroscience*, 2019, 20.
- Jain, S. M. (2022). Hugging Face. En *Introduction to Transformers for NLP*. Apress, Berkeley, CA.
- Jamsa, K. (2012). Cloud Computing: SaaS, PaaS, IaaS, Virtualization, Business Models, Mobile, Security and More. En *Jones & Bartlett Publishers*.
- Jannach, D., Pu, P., Ricci, F., & Zanker, M. (2022). Recommender systems: Trends and frontiers. *AI Magazine*, 43(2), 145-150.
- Jansen, B. J. (1998). The Graphical User Interface: An Introduction. *SIGCHI Bulletin*, 30(2), 22-26.
- Javed, M. (2020). Using Cosine Similarity to Build a Movie Recommendation System [Blog]. *Towards Data Science*. <https://towardsdatascience.com/using-cosine-similarity-to-build-a-movie-recommendation-system-ae7f20842599>
- Jayalakshmi, S., Ganesh, N., Čep, R., & Murugan, J. S. (2022). Movie Recommender Systems: Concepts, Methods, Challenges, and Future Directions. *Sensors*, 22, 4904.
- Jiao, A. (2020). An Intelligent Chatbot System Based on Entity Extraction Using RASA NLU and Neural Network. *Journal of Physics: Conference Series*, 1487(1), 012014.
- Joy, J. (2020). Deep Learning based Recommender Systems [Blog]. *Towards Data Science*. <https://towardsdatascience.com/deep-learning-based-recommender-systems-3d120201db7e>
- Koren, Y., Bell, R., & Volinsky, C. (2009). Matrix factorization techniques for recommender systems. *Computer*, 42(8), 30-37.

- Kouloumpis, E., Wilson, T., & Moore, J. (2011). Twitter Sentiment Analysis: The Good the Bad and the OMG! *Fifth International AAAI Conference on Web and Social Media*, 5(1), 538-541.
- Krukowski, I. (2022). How to translate languages in Python with Google Translate and DeepL (plus more) [Blog]. *Lokalise*. <https://lokalise.com/blog/how-to-translate-languages-in-python-with-google-translate-and-deepl-plus-more/>
- Kumar, A., Tejaswini, P., Nayak, O., Deep Kujur, A., Gupta, R., Rajanand, A., & Sahu, M. (2022). A Survey on IBM Watson and Its Services. *Journal of Physics: Conference Series*, 2273(1), 012022.
- Lazarus, R. S. (1991). *Emotion and adaptation*. Oxford University Press.
- LeDoux, J. E. (1998). *The emotional brain: The mysterious underpinnings of emotional life*. Simon & Schuster.
- Lin, C. C., Huang, A., & Yang, S. (2023). A Review of AI-Driven Conversational Chatbots Implementation Methodologies and Challenges (1999–2022). *Sustainability* 2023, 15(5), 4012.
- Liu, B. (2012). *Sentiment Analysis and Opinion Mining*. Springer Cham.
- Lops, P., de Gemmis, M., & Semeraro, G. (2011). Content-based Recommender Systems: State of the Art and Trends. *Recommender Systems Handbook*, 73-105.
- Maas, A., Daly, R. E., Pham, P. T., Huang, D., Ng, A. Y., & Potts, C. (2011). Learning Word Vectors for Sentiment Analysis. En *Forty-ninth Annual Meeting of the Association for Computational Linguistics* (pp. 142-150).
- Maeda, J., & Bolaños, M. (2023). Choosing an LLM Model. *Microsoft Learn*.
- Mahood, O. (2022). What's Hugging Face? [Blog]. *Towards Data Science*. <https://towardsdatascience.com/whats-hugging-face-122f4e7eb11a>
- Majumder, N., Poria, S., Hazarika, D., Mihalcea, R., Gelbukh, A., & Cambria, E. (2018). DialogueRNN: An Attentive RNN for Emotion Detection in Conversations. *Thirty-Third AAAI Conference on Artificial Intelligence*, 6818-6825.
- Marinescu, D. C. (2013). *Cloud Computing: Theory and Practice*. Newnes.

- Mbaabu, O. (2021). Recommender System Explained [Blog]. *section.io*.
<https://www.section.io/engineering-education/recommender-system-explained/>
- McTear, M., Callejas, Z., & Griol, D. (2016). *The conversational interface: Talking to smart devices*. Springer Cham.
- McTear, M. F. (1998). Modelling spoken dialogues with state transition diagrams: experiences with the CSLU toolkit. *Fifth International Conference on Spoken Language Processing*, 4.
- Medhat, W., Hassan, A., & Korashy, H. (2014). Sentiment analysis algorithms and applications: A survey. *Ain Shams Engineering Journal*, 5(4), 1093-1113.
- Melville, P., & Sindhvani, V. (2011). Recommender Systems. *Encyclopedia of Machine Learning*, 829-838.
- Metzinger, T., & Wiese, W. (2017). *Philosophy and Predictive Processing*. MIND Group.
- Meyer, P. (2023). *14 criteria for (well) choosing a chatbot solution*. Towards Data Science.
- Mooney, R. J., & Roy, L. (2000). Content-based book recommending using learning for text categorization. *ACM International Conference on Digital Libraries*, 195-204.
- Muhammad, A. F., Susanto, D., Alimudin, A., Adila, F., Assidiqi, M. H., & Nabhan, S. (2020). Developing English Conversation Chatbot Using Dialogflow. *2020 International Electronics Symposium*, 468-475.
- Nandwani, P., & Verma, R. (2021). A review on sentiment analysis and emotion detection from text. *Social Network Analysis and Mining*, 11(81), 1-19.
- Nielsen, F. Å. (2011). A new ANEW: Evaluation of a word list for sentiment analysis in microblogs. *ESWC2011 Workshop on «Making Sense of Microposts»*, 718, 93-98.
- Ochsner, K. N., & Gross, J. J. (2005). The cognitive control of emotion. *Trends in Cognitive Sciences*, 9(5), 242-249.
- Pal, A., Barigidad, A., & Mustafi, A. (2020). *IMDb Movie Reviews Dataset*. IEEE Dataport.
- Pang, B., & Lee, L. (2008). Opinion mining and sentiment analysis. *Foundations and Trends in Information Retrieval*, 2(1-2), 1-135.

- Pang, B., Lee, L., & Vaithyanathan, S. (2002). Thumbs up? Sentiment Classification using Machine Learning Techniques. *Conference on Empirical Methods in Natural Language Processing*, 79-86.
- Pardo, D. (2019). ¿Qué es y para qué sirve una API? Aquí las respuestas necesarias [Blog]. *Pandora FMS*. <https://pandorafms.com/blog/es/para-que-sirve-una-api/>
- Pazzani, M. J., & Billsus, D. (2007). Content-based recommendation systems. *The Adaptive Web*, 4321 LNCS, 325-341.
- Pearson, K. (1895). Note on Regression and Inheritance in the Case of Two Parents on JSTOR. *Royal Society of London*, 58, 240-242.
- Phan, K. L., Wager, T., Taylor, S. F., & Liberzon, I. (2002). Functional neuroanatomy of emotion: A meta-analysis of emotion activation studies in PET and fMRI. *NeuroImage*, 16(2), 331-348.
- Phelps, E. A. (2006). Emotion and cognition: insights from studies of the human amygdala. *Annual review of psychology*, 57, 27-53.
- Plutchik, R. (1980). *Emotion: A Psychoevolutionary Synthesis*. Harper & Row.
- Poria, S., Cambria, E., Bajpai, R., & Hussain, A. (2017). A review of affective computing: From unimodal analysis to multimodal fusion. *Information Fusion*, 37, 98-125.
- Radford, A., Narasimhan, K., Salimans, T., & Sutskever, I. (2018). Improving language understanding by generative pre-training. *OpenAI*, 12.
- Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., & Sutskever, I. (2019). Language Models are Unsupervised Multitask Learners. *OpenAI blog*, 1, 9.
- Radziwill, N. M., & Benton, M. C. (2017). Evaluating Quality of Chatbots and Intelligent Conversational Agents. *arXiv*, 21.
- Ramming, L., Konadl, D., & Leist, S. (2022). Chatbot Design Features to Increase Productivity. *International Conference on Design Science Research in Information Systems and Technology*, 6.
- Reimers, N., & Gurevych, I. (2019). Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. *2019 Conference on Empirical Methods in Natural Language*

Processing and Ninth International Joint Conference on Natural Language Processing, 3982-3992.

- Rendle, S., Freudenthaler, C., Gantner, Z., & Schmidt-Thieme, L. (2012). BPR: Bayesian Personalized Ranking from Implicit Feedback. *Twenty-fifth Conference on Uncertainty in Artificial Intelligence*, 452-461.
- Resnick, P., Iacovou, N., Suchak, M., Bergstrom, P., & Riedl, J. (1994). GroupLens: An open architecture for collaborative filtering of netnews. *1994 ACM Conference on Computer Supported Cooperative Work*, 175-186.
- Ricci, F., Rokach, L., & Shapira, B. (2011). Introduction to Recommender Systems Handbook. *Recommender Systems Handbook*, 1-35.
- Roy, D., & Dutta, M. (2022). A systematic review and research perspective on recommender systems. *Journal of Big Data*, 9(1), 1-36.
- Ruder, S., Peters, M. E., Swayamdipta, S., & Wolf, T. (2019). Transfer Learning in Natural Language Processing. *NAACL HLT 2019*, 15-18.
- Salakhutdinov, R., Mnih, A., & Hinton, G. (2007). Restricted Boltzmann machines for collaborative filtering. *Twenty-fourth International Conference on Machine Learning*, 227, 791-798.
- Sanh, V., Debut, L., Chaumond, J., & Wolf, T. (2019). DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. *arXiv*, 5.
- Saravia, E., Toby Liu, H. C., Huang, Y. H., Wu, J., & Chen, Y. S. (2018). CARER: Contextualized Affect Representations for Emotion Recognition. *2018 Conference on Empirical Methods in Natural Language Processing*, 3687-3697.
- Sarwar, B., Karypis, G., Konstan, J., & Riedl, J. (2001). Item-based collaborative filtering recommendation algorithms. *Tenth International Conference on World Wide Web*, 285-295.
- Sebastian, G. (2023). Privacy and Data Protection in ChatGPT and Other AI Chatbots: Strategies for Securing User Information. *International Journal of Security and Privacy in Pervasive Computing*, 15(1), 1-14.

- Shawar, B., & Atwell, E. (2007). Chatbots: are they really useful? *LDV Forum 2007*, 22(1), 29-49.
- Socher, R., Perelygin, A., Wu, J., Chuang, J., Manning, C. D., Ng, A. Y., & Potts, C. (2013). Recursive Deep Models for Semantic Compositionality Over a Sentiment Treebank. *2013 Conference on Empirical Methods in Natural Language Processing*, 1631-1642.
- Srinivasan, A. (2019). Recommendation Systems using Reinforcement Learning. *IBM Data Science in Practice*.
- Strapparava, C., & Mihalcea, R. (2007). SemEval-2007 Task 14: Affective Text. *Fourth International Workshop on Semantic Evaluations*, 70-74.
- Su, X., & Khoshgoftaar, T. M. (2009). A Survey of Collaborative Filtering Techniques. *Advances in Artificial Intelligence*, 2009, 1-19.
- Tang, Y. Y., Hölzel, B. K., & Posner, M. I. (2015). The neuroscience of mindfulness meditation. *Nature Reviews Neuroscience*, 16(4), 213-225.
- Teslenko, N. (2023). What Are Transformers in NLP: Benefits and Drawbacks. *Pangeanic*.
- Topsakal, O., & Akinci, T. C. (2023). Creating Large Language Model Applications Utilizing LangChain: A Primer on Developing LLM Apps Fast. *Fifth International Conference on Applied Engineering and Natural Sciences*, 1(1), 1050-1056.
- Tracy, J. L., & Randles, D. (2011). Four Models of Basic Emotions: A Review of Ekman and Cordaro, Izard, Levenson, and Panksepp and Watt. *Emotion Review*, 3(4), 397-405.
- Tunstall, L., Von Werra, L., & Wolf, T. (2022). *Natural Language Processing with Transformers (Revised Edition)*. O'Reilly Media.
- Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention Is All You Need. *Thirty-First International Conference on Neural Information Processing Systems*, 6000-6010.
- Wali, E., Chen, Y., Mahoney, C. M., Middleton, T. G., Babaeianjelodar, M., Njie, M., & Matthews, J. N. (2020). Is Machine Learning Speaking my Language? A Critical Look at the NLP-Pipeline Across 8 Human Languages. *ArXiv*, 7.

- Wei, J., Kim, S., Jung, H., & Kim, Y.-H. (2023). Leveraging Large Language Models to Power Chatbots for Collecting User Self-Reported Data. *arXiv*, 22.
- Weizenbaum, J. (1966). ELIZA - a computer program for the study of natural language communication between man and machine. *Communications of the ACM*, 9(1), 36-45.
- Westcott, K., Arbanas, J., Arkenberg, C., & Auxler, B. (2021). SVOD, social media, and gaming trends. *Deloitte*.
- William, P., Lanke, G. R., Inukollu, V. N. R., Singh, P., Shrivastava, A., & Kumar, R. (2023). Framework for Design and Implementation of Chat Support System using Natural Language Processing. *2023 fourth International Conference on Intelligent Engineering and Management*, 1-7.
- Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M., Davison, J., Shleifer, S., Platen, P. von, Ma, C., Jernite, Y., Plu, J., Xu, C., Scao, T. Le, Gugger, S., ... Rush, A. M. (2020). Transformers: State-of-the-Art Natural Language Processing. *2020 EMNLP*, 38-45.
- World Wide Web Consortium. (2017). Cascading Style Sheets. *World Wide Web Consortium (W3C)*.
- World Wide Web Consortium. (2023). HTML Standard. *World Wide Web Consortium (W3C)*.
- Yuyan, Z., Xiayao, S., & Yong, L. (2019). A Novel Movie Recommendation System Based on Deep Reinforcement Learning with Prioritized Experience Replay. *2019 IEEE nineteenth International Conference on Communication Technology*, 1496-1500.
- Zhang, L., Wang, S., & Liu, B. (2018). Deep Learning for Sentiment Analysis : A Survey. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 8(4), e1253.
- Zhang, Z., Han, X., Liu, Z., Jiang, X., Sun, M., & Liu, Q. (2019). ERNIE: Enhanced Language Representation with Informative Entities. *Fifty-seventh Annual Meeting of the Association for Computational Linguistics*, 1441-1451.

Anexo A. Artículo de investigación

Sistema de recomendación de películas basado en análisis emocional con chatbot, AngellA.

Rubén Aponte Núñez, Edison Giraldo Pérez, Joel Orellana Martínez

Universidad Internacional de la Rioja, Logroño (España)

20 de septiembre de 2023

RESUMEN

La presente investigación se enfoca en el diseño de un sistema de recomendación de películas basado en el análisis de las emociones, implementando un sistema de conversación interactivo llamado AngellA. Este tiene como principal objetivo optimizar la experiencia del usuario, proponiendo películas que sean acordes a sus estados emocionales predominantes. La metodología adoptada para este proyecto se estructura en distintas etapas: la recopilación de datos cinematográficos, el desarrollo y entrenamiento de un modelo especializado para el análisis de las emociones y la configuración y despliegue del sistema de recomendaciones. Los resultados muestran una capacidad notable de clasificar con precisión las emociones, lo que resulta en recomendaciones cinematográficas pertinentes. Por su parte, las evaluaciones ponen de manifiesto la habilidad del sistema para entregar sugerencias que estén acordes con el perfil emocional del usuario. Es evidente que la aplicación del análisis de las emociones en este contexto potencia de manera significativa la personalización de las recomendaciones, lo que a su vez conduce a un aumento en la satisfacción del usuario.

PALABRAS CLAVE

Análisis emocional, Chatbot, Sistemas de recomendación, Películas, Procesamiento del lenguaje natural

I. INTRODUCCIÓN

En este artículo, se propone un sistema de recomendación de películas basado en análisis emocional integrado en un chatbot [1], llamado "AngellA", la cual simula mantener una conversación con una persona al proveer recomendaciones automáticas, utilizando técnicas avanzadas de Procesamiento del Lenguaje Natural (NLP, por sus siglas en inglés), como lo son los actualmente muy utilizados transformers [2]. Además, se considera el análisis emocional para ofrecer recomendaciones de películas más precisas y personalizadas.

La introducción de la arquitectura de transformers en el sistema de recomendación permite una mayor precisión y adaptabilidad en la captura de las preferencias y emociones de los usuarios en tiempo real, dado que estos utilizan mecanismos de autoatención para capturar dependencias a largo plazo en secuencias de texto y han demostrado ser eficientes en una amplia gama de tareas de NLP, incluida la comprensión y generación de texto [2]. Los transformers han logrado el entrenamiento masivo de corpus de texto, los Grandes Modelos de Lenguaje (Large Language Models o LLM) capaces de poder realizar una amplia gama de tareas de NLP como la generación de texto, por ejemplo, entrenar un chatbot a través de una instrucción (Prompt en inglés) en lenguaje natural [3].

II. ESTADO DEL ARTE

2.1. Avances en el estudio de emociones

En la década de 1980, la investigación de las emociones tomó un giro significativo con la formulación de la Teoría Psicoevolucionista de las Emociones [4]. Esta teoría,

basada en la idea que las emociones desempeñan un papel adaptativo en la evolución humana, postula que sirven como mecanismos adaptativos en respuesta a los desafíos del entorno. Uno de sus logros más notables fue la identificación de ocho emociones primarias organizadas en pares opuestos, como alegría versus tristeza, confianza versus aversión, sorpresa versus anticipación, y enojo versus temor.

Otra contribución importante fue la creación de la "rueda de las emociones", una representación gráfica que ilustra la estructura y jerarquía de las emociones [4]. Esta rueda no solo destaca las emociones primarias y sus opuestos, sino que también permite visualizar las combinaciones y grados de intensidad de cada emoción, como se muestra en la **Figura 1**.

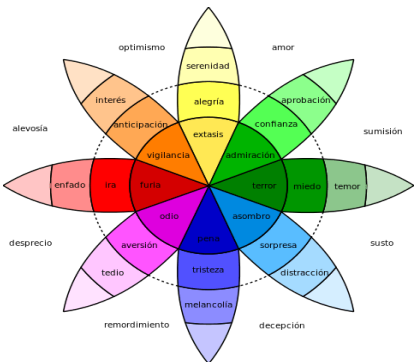


Figura 1. Rueda de las emociones.

2.2. Avances en la detección de sentimientos y emociones en texto mediante IA

Los primeros esfuerzos en la detección de emociones en texto se centraron en el análisis de sentimiento, buscando la capacidad de identificar si un fragmento de texto tiene una carga emocional positiva, negativa o neutra. Uno de los hitos tempranos fue la creación del Diccionario de Sentimiento AFINN [5], que asigna puntuaciones de sentimiento a palabras en inglés.

Otro hito significativo en esta área tuvo lugar en 2013 con la presentación del enfoque basado en Redes Neuronales Recurrentes (RNN, por sus siglas en inglés). Este enfoque contribuyó de manera fundamental a una comprensión más profunda y contextual de las emociones presentes en el texto [6].

Posteriormente la incorporación de modelos de atención y contexto en la detección de emociones marcó el estándar de la industria. Estos modelos consideran no solo las palabras individuales en el texto, sino también las relaciones y conexiones entre ellas. Un ejemplo es el *Transformer*, un modelo de atención múltiple, que ha demostrado ser efectivo en el análisis de texto emocional [2], la evolución de este modelo a otros como BERT [7] y GPT [8] hicieron una revolución en la detección de sentimientos y emociones. Estos modelos, entrenados en vastas cantidades de texto, pueden transferir su conocimiento previo a tareas específicas de análisis de emociones, mejorando la precisión y la capacidad de adaptación a diferentes dominios.

2.3. Enfoques en la extracción de emociones

Léxico: utilizan diccionarios y listas de palabras previamente construidas y asociadas con sentimientos o emociones específicas [9], [10]. Estos enfoques suelen ser simples y rápidos, pero pueden ser limitados en términos de cobertura y flexibilidad.

Machine learning: emplean técnicas de clasificación supervisada, como máquinas de vectores de soporte (SVM, por sus siglas en inglés), árboles de decisión y clasificadores Bayesianos, para predecir sentimientos y emociones a partir de características extraídas del texto [11][12]. Aunque estos métodos pueden ofrecer un rendimiento mejor que los enfoques basados en léxicos, requieren datos etiquetados y pueden ser sensibles a la selección de características y la elección del algoritmo.

Deep learning: las redes neuronales convolucionales [13] (CNN, por sus siglas en inglés) y las RNN [14], han demostrado ser efectivas en el análisis de sentimientos y la detección de emociones debido a su capacidad para modelar representaciones de texto de alto nivel y capturar información contextual [6], [15]. Sin embargo, estos enfoques pueden requerir grandes cantidades de datos y recursos computacionales para entrenamiento y ajuste de modelos. [16].

2.4. Sistemas de recomendación

Los sistemas de recomendación son herramientas esenciales en diversas aplicaciones, como comercio electrónico, entretenimiento y redes sociales, para ayudar a los usuarios a descubrir contenido relevante y personalizado [17]. En su esencia, es un sistema que emplea algoritmos para sugerir elementos o productos a usuarios cuando visitan un sitio web o exploran un producto en línea. Estas sugerencias pueden abarcar diversas categorías, desde productos que visualizar y adquirir hasta películas que ver, anuncios para visualizar e

información para leer.

La taxonomía propuesta en *Practical Recommender Systems* [18], describe un sistema en función de varias dimensiones como dominio, propósito, contexto, nivel de personalización, opiniones, privacidad y confiabilidad, interfaces y algoritmos mediante los cuales puede describirse por completo un sistema de recomendación.

2.5. *Chatbots* en sistemas de recomendación

La incorporación de *chatbots* en sistemas de recomendación ha ganado popularidad en los últimos años debido a su capacidad para interactuar con los usuarios de manera natural y en tiempo real [19]. Los *chatbots* pueden mejorar la experiencia de usuario y facilitar la exploración y el descubrimiento de contenido, al tiempo que proporcionan información adicional y responden a preguntas relacionadas con el contenido recomendado [20].

Chatbots basados en reglas: los *chatbots*, desde su concepción inicial, han evolucionado notablemente en su complejidad y aplicaciones. Uno de los primeros enfoques en la construcción de *chatbots* fue el modelo basado en reglas. Estos *chatbots* funcionan mediante un conjunto predefinido de reglas que dictan cómo el *chatbot* debe responder ante entradas específicas de los usuarios [21]. Dicho de otro modo, para cada entrada específica del usuario, existe una respuesta predefinida por parte del *chatbot*. El *chatbot* ELIZA, creado por Joseph Weizenbaum en el *Massachusetts Institute of Technology* (MIT), es uno de los ejemplos más emblemáticos de este tipo de sistema, simulando una terapeuta empática y proporcionando respuestas basadas en scripts preprogramados.

Chatbots basados en técnicas de DL: el uso de técnicas avanzadas de NLP, como los *transformers* en *chatbots* ha sido un área de investigación activa [2]. Los *transformers* han demostrado ser efectivos en una amplia gama de tareas de NLP, incluido el análisis de sentimientos, la respuesta a preguntas y la generación de texto, gracias a su capacidad para capturar información contextual a largo plazo y modelar representaciones semánticas de alto nivel [7] [22].

Recientemente, los LLM con billones de parámetros pre entrenados en *corpus* gigantescos de texto (GPT, Llama, PALM, etc.) [3] han demostrado habilidades para generar respuestas coherentes y contextuales, mejorando significativamente la calidad y la naturalidad de las interacciones con los usuarios.

III. OBJETIVOS Y METODOLOGÍA

A. Objetivo general

Desarrollar un *chatbot* para recomendación de películas personalizadas que utilice enfoques avanzados de NLP, que logre una comprensión y análisis de las emociones y preferencias del usuario, proporcionando recomendaciones altamente relevantes y precisas.

B. Objetivos específicos

1. Investigar y analizar las técnicas y algoritmos de NLP más adecuados para el análisis de emociones.
2. Revisar y comparar diferentes enfoques y arquitecturas de sistemas de recomendación de películas, incluidos los métodos basados en contenido, filtrado colaborativo y enfoques híbridos.

3. Desarrollar e implementar un modelo de clasificación de emociones para películas, basado en la información textual y las opiniones de los usuarios.
4. Diseñar e implementar un *chatbot* que utilice técnicas de NLP y el modelo de clasificación de emociones para generar recomendaciones de películas basadas en el estado de ánimo y los gustos del usuario.
5. Evaluar la efectividad y precisión del *chatbot* para recomendar películas mediante pruebas con usuarios y análisis de sus respuestas y satisfacción en función de las recomendaciones recibidas.

C. Metodología del trabajo

A continuación, se describe la metodología del trabajo:

Revisión bibliográfica: en esta etapa se llevó a cabo una revisión exhaustiva de la literatura científica y académica relacionada con NLP, el análisis de sentimientos, los sistemas de recomendación de películas y los *chatbots*.

Creación del modelo de clasificación de emociones: en esta etapa, se desarrolló e implementó un modelo de clasificación de emociones para películas, utilizando la información textual y las opiniones de los usuarios.

Diseño e implementación del *chatbot*: con el modelo de clasificación de emociones desarrollado en la etapa anterior, se diseñó e implementó un *chatbot*.

Evaluación del *chatbot* para recomendar películas: en esta etapa, se llevará a cabo la evaluación de la efectividad y precisión del *chatbot* para recomendar películas.

Análisis de resultados y conclusiones: por último, se analizarán los resultados obtenidos.

IV. CONTRIBUCIÓN

La adecuada preparación de un conjunto de datos constituyó un paso esencial en el proceso de construcción de un modelo clasificatorio destinado a identificar las emociones percibidas en películas por consumidores previos. Para el desarrollo del presente estudio se seleccionó el *conjunto de datos IMDB – IEEE Movie reviews* [23]: que incluye casi 1 millón de reseñas provenientes de 1.150 películas distintas registradas en IMDb y 17 géneros distintos.

4.1. Creación del modelo para el análisis de emociones

Para abordar la tarea de clasificación emocional en las respuestas de los usuarios y las reseñas de películas, se optó por aprovechar la potencia de los modelos pre entrenados en NLP. Uno de estos modelos, en particular, el "*distilbert-base-uncased*", mostró ser una elección adecuada debido a su eficiencia y rendimiento en tareas de clasificación textual [24].

Al emplear el modelo "*distilbert-base-uncased*", previamente entrenado en una amplia gama de textos, se aprovechó su capacidad para comprender el contexto y la semántica del lenguaje. Este modelo tiene la capacidad de captar matices emocionales sutiles presentes en el lenguaje humano, lo que fue crucial para el éxito de la tarea de clasificación emocional en este proyecto (ver *Figura 2*).

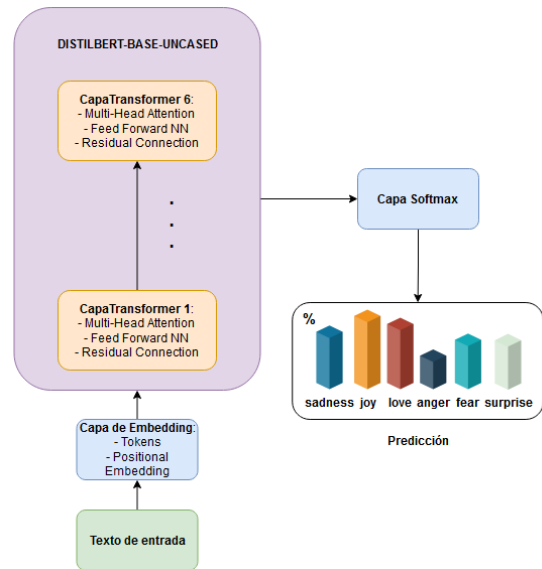


Figura 2. Modelo "*distilbert-base-uncased*" con *fine tuning* para clasificación de emociones.

4.1.1. Entrenamiento del modelo

Para el entrenamiento del modelo se utilizó el *pipeline* de *Hugging Face* y *PyTorch* y se configuraron los parámetros de entrenamiento, como el tamaño del lote (*batch_size*), la frecuencia de registro de métricas (*logging_steps*), el número de epochs de entrenamiento (*num_train_epochs*), la tasa de aprendizaje (*learning_rate*), entre otros. Los parámetros más relevantes de configuración del modelo se presentan en la *Tabla 1*.

Tabla 1. Parámetros de entrenamiento del modelo de clasificación de emociones.

Parámetro	Valor
<i>model_name</i>	<i>distilbert-base-uncased-finetuned-417k</i>
<i>batch_size</i>	64
<i>num_train_epochs</i>	2
<i>learning_rate</i>	2.5×10^{-5}
<i>weight_decay</i>	0.01

El proceso de entrenamiento tuvo una duración de 44 minutos en 2 *epochs* y se obtuvieron las métricas presentes en la *Tabla 2*.

Tabla 2. Resultado de Métricas del modelo clasificador de emociones.

Dataset	Training Loss	Test Loss	Accuracy	F1 Score
Train	0.0802	-	0.9448	0.9426
Test	-	0.8843	0.9392	0.9385

Para la evaluación de modelo de clasificación, se utilizó la matriz de confusión que permite visualizar el desempeño del modelo para cada clase específica.

En la **Figura 3**, se muestra la matriz de confusión del modelo. Las filas de la matriz representan las emociones reales, mientras que las columnas reflejan las predicciones del modelo. Se puede observar que la matriz de confusión muestra que el modelo es más preciso a la hora de detectar tristeza (99%), seguida de alegría (98%). El modelo es menos preciso a la hora de detectar la ira, el miedo, la sorpresa y el amor. También hay algunos falsos positivos y falsos negativos en la matriz de confusión.

Estos resultados sugieren que el modelo está funcionando adecuadamente en la clasificación de emociones, pero todavía hay margen de mejora, especialmente en la detección de emociones menos comunes como la sorpresa y el amor. Trabajos futuros podrían centrarse en mejorar la precisión del modelo a la hora de detectar estas emociones.

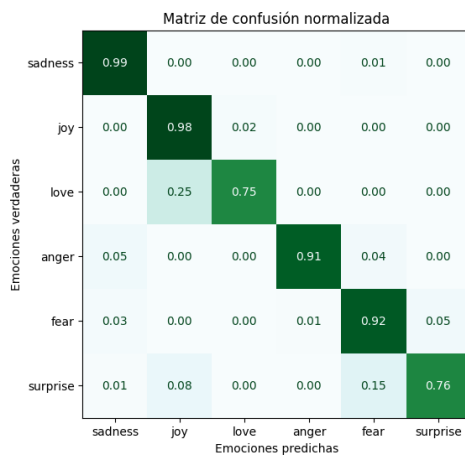


Figura 3. Matriz de confusión para evaluar la precisión del modelo de clasificación de emociones.

El modelo fue subido a un repositorio de Hugging Face creado para este trabajo llamado: *TFMUNIR/distilbert-base-uncased-finetuned-417k*.

4.1.2. Segmentación del conjunto de datos de películas

Se realizó una segmentación del conjunto de datos debido a la necesidad de incrementar la variabilidad y distribución en las recomendaciones, garantizando un enfoque más dirigido y personalizado. Al dividir el conjunto de datos en grupos homogéneos basados en criterios como género, año o emociones, se posibilitó una personalización más afinada de las recomendaciones, alineándolas con las emociones del usuario en un momento específico.

Se estableció la necesidad de analizar un espectro de segmentos que abarca desde 1 hasta 30 segmentos tal como se observa en la **Figura 4**. A partir de esta exploración, se observó que un punto óptimo que proporciona una segmentación diversificada y representativa de las películas se sitúa alrededor de los 10 segmentos. Esta elección se alinea con un equilibrio entre la especificidad y la generalidad: se distancia del valor identificado por el método del codo, que sugiere $k = 5$, evitando una posible subsegmentación, pero a su vez, no se inclina hacia valores excesivamente elevados, superiores a $k = 10$, que podrían conllevar una sobresegmentación y, por ende, reducir la robustez y

utilidad práctica de los segmentos definidos.

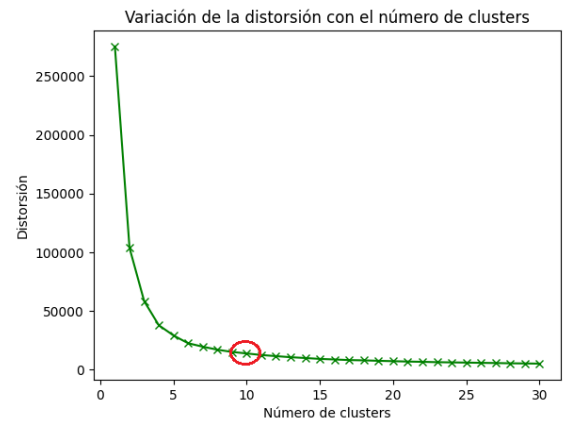


Figura 4. Variación de la distorsión con el número de clusters.

Para $k = 10$, la segmentación de las películas se puede observar en la **Figura 5**.

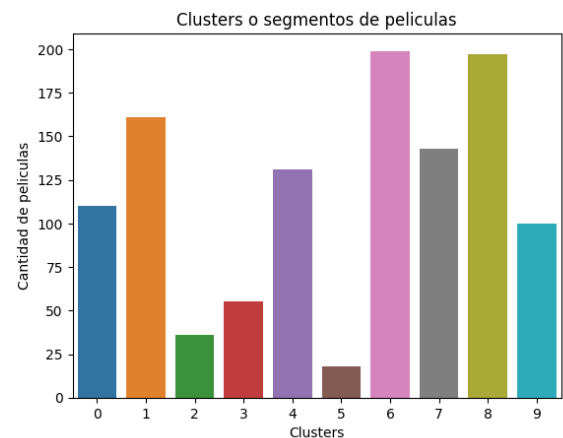


Figura 5. Variación de la distorsión con el número de clusters.

4.1.3. Reajuste emocional para recomendación de películas

Con base en la rueda de las emociones presentada en la Sección 1, se desarrolló un modelo que reajusta emociones negativas detectadas en las interacciones del usuario con el *chatbot*, orientándolas hacia sus contrapartes positivas. El objetivo de este ajuste es que, al identificar emociones tales como el miedo, la tristeza o el enojo, estas sean procesadas por un algoritmo que, sin eliminar la emoción negativa original, pondere hacia su opuesto positivo.

Esta metodología tiene como finalidad ofrecer recomendaciones de películas que sean más amenas y placenteras, con el propósito de contribuir a la mejora del estado anímico del usuario. Esta estrategia se alinea con la idea de proporcionar una experiencia de usuario optimizada y empática, reconociendo y respondiendo adecuadamente a sus estados emocionales. La propuesta de balance de emociones se muestra en la **Tabla 3**.

Tabla 3. Contrapartes de emociones implementadas en el *chatbot*.

Emoción del usuario	Contraparte emocional del modelo
Tristeza (sadness)	Alegría (joy)
Alegría (joy)	Alegría (joy)
Amor (love)	Amor (love)
Rabia (anger)	Amor (love)
Temor (fear)	Alegría (joy)
Sorpres (surprise)	Alegría (joy)

En la implementación del modelo, específicamente si una emoción como la tristeza, excede un umbral definido (en este caso 0.2), el modelo incrementa su contraparte positiva (alegría) mientras reduce proporcionalmente la emoción original. Esta operación se realiza mediante un factor de ajuste, que en este contexto se ha establecido en 0.5. Posteriormente, para garantizar una respuesta coherente, el vector resultante es normalizado, de modo que la suma total de sus componentes emocionales o probabilidades sea 1. De esta manera, el sistema proporciona una respuesta emocionalmente equilibrada y adaptada al contexto del usuario.

4.2. Desarrollo del sistema de recomendación integrado en el *chatbot*

4.2.1. *Chatbot*: AngellIA

"AngellIA" es un sistema de IA diseñado para brindar a los usuarios recomendaciones de películas personalizadas a través de interacciones en tiempo real. Para lograr esta funcionalidad, el sistema se apoya en un *backend* desarrollado en el *framework Django*, que comprende una variedad de funciones y vistas para llevar a cabo tareas esenciales como procesamiento de texto, análisis de emociones, recomendación de películas e interacción con el usuario. Además, el sistema se completa con un *frontend* desarrollado en HTML, JavaScript y CSS, que brinda la interfaz con la que los usuarios interactúan.

El diagrama de la **Figura 6** presenta el funcionamiento simplificado de AngellIA.

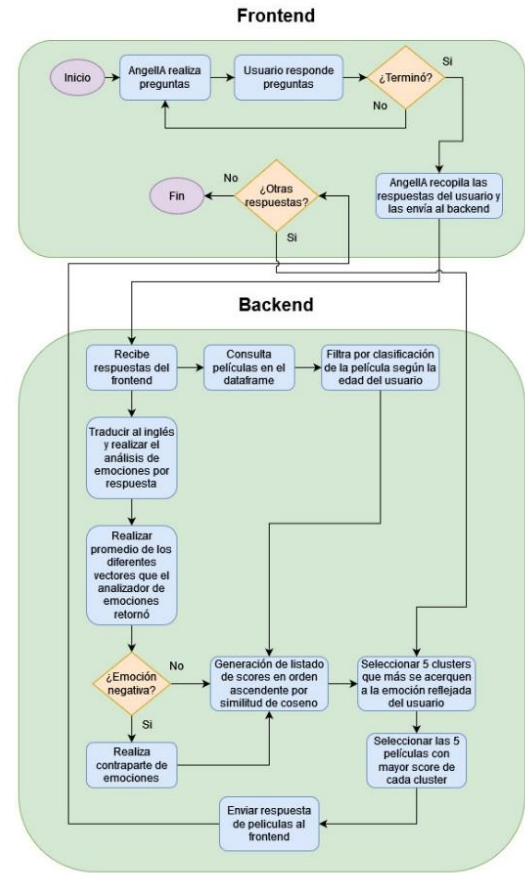


Figura 6. Funcionamiento simplificado de AngellIA.

4.2.2. *Backend*

El backend coordina las operaciones para ofrecer recomendaciones cinematográficas personalizadas. Su estructura incluye un análisis de emociones, donde un modelo pre entrenado identifica y clasifica las emociones en los mensajes de los usuarios. En el procesamiento textual, también se emplean técnicas de NLP para analizar el contenido lingüístico del mensaje.

4.2.3. *Frontend*

La página web presenta un diseño sencillo, incluyendo un encabezado y un área donde se desarrolla la interacción entre el usuario y el chatbot, AngellIA. Inicialmente, el chatbot pregunta por la edad del usuario, dato esencial para filtrar recomendaciones. Posteriormente, AngellIA plantea interrogantes sobre el día del usuario, su estado anímico y eventos recientes, con el propósito de extraer emociones y sentimientos. Las preguntas buscan capturar una visión general del estado emocional, solicitar una representación narrativa de los sentimientos y obtener relatos de experiencias recientes. Una vez recabadas las respuestas, estas se almacenan en el backend para su posterior análisis y generación de recomendaciones cinematográficas. Al final, el usuario tiene la opción de solicitar más recomendaciones o finalizar la conversación. La interfaz de usuario puede verse en la **Figura 7**.



Figura 7. Frontend del chatbot AngellIA.

Cuando AngellIA recopila la información de los usuarios, genera recomendaciones de películas basadas en esas respuestas.

El usuario puede calificar las recomendaciones dando "Me gusta" o "No me gusta". Estas calificaciones se guardan en una base de datos *Postgres* para posterior análisis.

V. EVALUACIÓN Y RESULTADOS

5.1. Pruebas del sistema

Las pruebas realizadas buscaron evaluar la pertinencia de las recomendaciones del sistema en distintos contextos emocionales. Las recomendaciones dirigidas a emociones negativas se inclinaban hacia géneros como drama, romance, aventura, suspenso y ciencia ficción, que tratan temas de superación y reafirmación de vida. Las películas sugeridas presentan tramas que permiten al usuario confrontar y procesar sus emociones. Por otro lado, en respuesta a emociones positivas, el sistema prioriza géneros como comedia, aventura, animación y acción, aunque mantiene una diversidad en sus sugerencias, permitiendo la exploración de nuevas temáticas por parte del usuario.

5.2. Pruebas realizadas por terceros

En el marco del presente estudio, se llevaron a cabo pruebas promocionando el sitio web AngellIA a través de diversas plataformas de redes sociales. Durante un período comprendido entre el 20 y el 28 de agosto de 2023, se extendió una invitación a potenciales usuarios para que procedieran a registrarse y explorar el funcionamiento del sistema. Es importante mencionar que, dentro de la base de datos, únicamente quedaron registrados aquellos usuarios que efectuaron evaluaciones sobre las recomendaciones presentadas. Como resultado de esta fase, se contabilizaron 183 calificaciones aportadas por un

conjunto de 68 usuarios diferentes.

La **Figura 8** muestra que ante una emoción predominante positiva no realiza un reajuste del vector de emociones del usuario y pasa a la etapa de identificación de la película por medio de la distancia de coseno más cercana, que en este caso arroja un valor de similitud de 0.99. El usuario manifestó que la recomendación fue de su agrado.

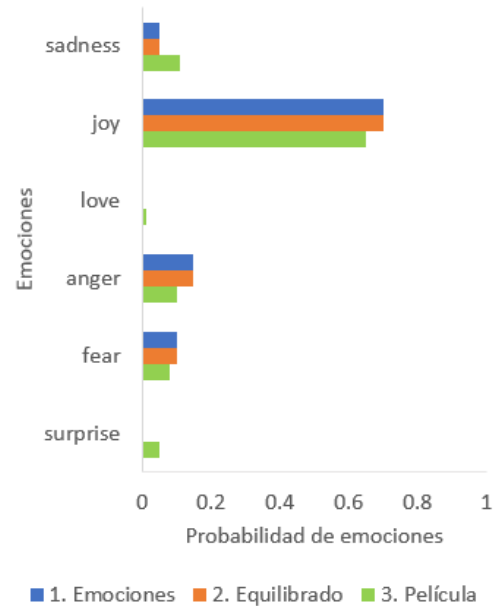


Figura 8. Recomendación de película con emoción del usuario altamente positiva.

La **Figura 9** ilustra una situación en la que el usuario experimenta una emoción combinada. El sistema modifica el vector de emociones del usuario, reduciendo la intensidad de las emociones negativas como tristeza, rabia y temor, y potenciando las positivas, específicamente alegría y amor. Luego, el sistema avanza a la fase de identificación de la película utilizando la distancia de coseno, donde se obtuvo una similitud de 0.88. De acuerdo con la retroalimentación del usuario, la sugerencia fue satisfactoria.

Según se puede observar en la **Figura 10**, ante la presencia de una emoción predominantemente negativa, el sistema recalibra el vector emocional del usuario, minimizando las emociones negativas (rabia y tristeza) y amplificando las positivas (alegría y amor). Una vez realizado este ajuste, la elección de la película se realiza a través del método de distancia de coseno, arrojando una similitud de 0.74 en este caso particular. La validación por parte del usuario indica que la elección propuesta fue coherente con sus expectativas.

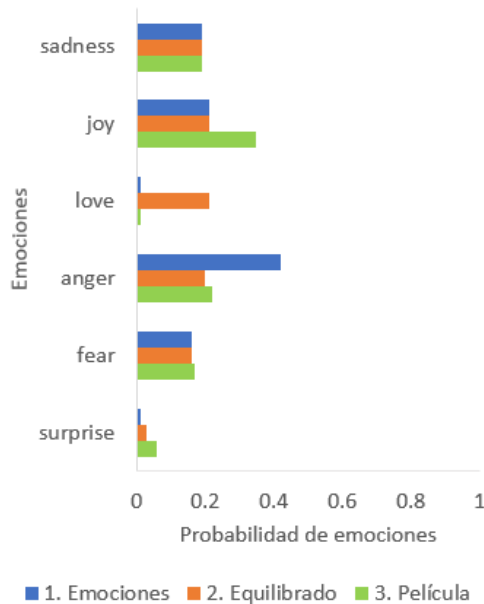


Figura 9. Recomendación de película con emociones mixtas del usuario.

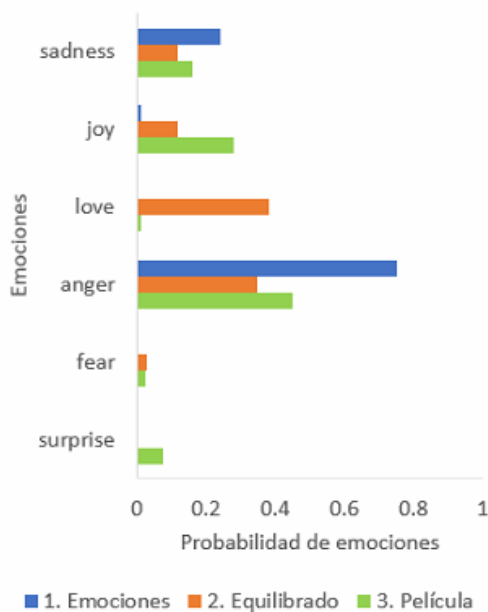


Figura 10. Recomendación de película con emoción del usuario altamente negativa.

VI. DISCUSIÓN

En base a los resultados obtenidos puede evidenciarse algunos aspectos relevantes:

1. El modelo entrenado para la detección de emociones en texto ha proporcionado resultados satisfactorios en la matriz de confusión que han permitido clasificar estados de ánimo de las personas de forma muy eficaz.
2. La detección de las emociones inherentes a las reseñas de consumidores previos de películas junto a la clusterización de ellas ha permitido presentar recomendaciones novedosas que dependen exclusivamente de las opiniones de los usuarios.
3. Ante la presencia de emociones particularmente

negativas, el sistema recomienda películas orientadas a mejorar el estado de ánimo.

VII. CONCLUSIONES

Las contribuciones de este trabajo se reflejan en la creación de un conjunto de datos de películas específicamente orientado a las emociones detectadas por espectadores previos y a las características de cada una de ellas (año, género, clasificación y sinopsis), así como la implementación de un modelo de análisis emocional y la construcción del sistema de recomendación integrado en el chatbot. Estos componentes se han diseñado para ofrecer recomendaciones de películas que se adapten mejor a las emociones de los usuarios, mejorando así la personalización y la satisfacción de estos.

La técnica de clusterización permitió agrupar películas con distintas variables sujetas a consideración, una de las cuales era el vector promedio de emociones de las reseñas realizadas por espectadores previos de las mismas, este enfoque innovador logró determinar un perfil emocional para cada película, lo que permitió al chatbot ofrecer recomendaciones a los usuarios basadas en la concordancia entre sus emociones y las evocadas por las películas.

En términos de evaluación del modelo de detección de emociones, se entrenó exitosamente un modelo de Transformers con alta precisión capaz de clasificar adecuadamente la mayoría de los estados de ánimo que fueron sujetos a estudio, como lo demostraron las métricas del modelo presentadas en la sección correspondiente.

La evaluación realizada al sistema de recomendación ha demostrado pertinencia en la adaptación de las recomendaciones de películas a las emociones de los usuarios. Los resultados obtenidos en las pruebas funcionales internas y de usuario (terceros) respaldan la eficacia de la solución propuesta.

REFERENCIAS

- [1] M. Dahiya, "A Tool of Conversation: Chatbot," *International Journal of Computer Sciences and Engineering*, vol. 5, no. 5, pp. 158–161, 2017.
- [2] A. Vaswani *et al.*, "Attention Is All You Need," *Thirty-First International Conference on Neural Information Processing Systems*, pp. 6000–6010, Jun. 2017.
- [3] J. Wei, S. Kim, H. Jung, and Y.-H. Kim, "Leveraging Large Language Models to Power Chatbots for Collecting User Self-Reported Data," *ArXiv*, p. 22, Jan. 2023.
- [4] R. Plutchik, *Emotion: A Psychoevolutionary Synthesis*. Harper & Row, 1980.
- [5] F. Å. Nielsen, "A new ANEW: Evaluation of a word list for sentiment analysis in microblogs," *ESWC2011 Workshop on "Making Sense of Microposts"*, vol. 718, pp. 93–98, Mar. 2011.

- [6] R. Socher *et al.*, “Recursive Deep Models for Semantic Compositionality Over a Sentiment Treebank,” *2013 Conference on Empirical Methods in Natural Language Processing*, pp. 1631–1642, 2013.
- [7] J. Devlin, M. W. Chang, K. Lee, and K. Toutanova, “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding,” *ArXiv*, p. 16, May 2019.
- [8] A. Radford, K. Narasimhan, T. Salimans, and I. Sutskever, “Improving language understanding by generative pre-training,” *OpenAI*, p. 12, 2018.
- [9] M. Hu and B. Liu, “Mining and summarizing customer reviews,” *Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 168–177, 2004.
- [10] C. Strapparava and R. Mihalcea, “SemEval-2007 Task 14: Affective Text,” *Fourth International Workshop on Semantic Evaluations*, pp. 70–74, 2007.
- [11] B. Pang, L. Lee, and S. Vaithyanathan, “Thumbs up? Sentiment Classification using Machine Learning Techniques,” *Conference on Empirical Methods in Natural Language Processing*, pp. 79–86, 2002.
- [12] E. Kouloumpis, T. Wilson, and J. Moore, “Twitter Sentiment Analysis: The Good the Bad and the OMG!,” *Fifth International AAAI Conference on Web and Social Media*, vol. 5, no. 1, pp. 538–541, Aug. 2011.
- [13] S. K. Bharti *et al.*, “Text-Based Emotion Recognition Using Deep Learning Approach,” *Comput Intell Neurosci*, vol. 2022, p. 8, 2022.
- [14] N. Majumder, S. Poria, D. Hazarika, R. Mihalcea, A. Gelbukh, and E. Cambria, “DialogueRNN: An Attentive RNN for Emotion Detection in Conversations,” *Thirty-Third AAAI Conference on Artificial Intelligence*, pp. 6818–6825, Nov. 2018.
- [15] Y. Y. Tang, B. K. Hölzel, and M. I. Posner, “The neuroscience of mindfulness meditation,” *Nat Rev Neurosci*, vol. 16, no. 4, pp. 213–225, Mar. 2015.
- [16] L. Zhang, S. Wang, and B. Liu, “Deep Learning for Sentiment Analysis : A Survey,” *Wiley Interdiscip Rev Data Min Knowl Discov*, vol. 8, no. 4, p. e1253, Jan. 2018.
- [17] F. Ricci, L. Rokach, and B. Shapira, “Introduction to Recommender Systems Handbook,” *Recommender Systems Handbook*, pp. 1–35, 2011.
- [18] K. Falk, *Practical Recommender Systems*. 2019.
- [19] D. Jannach, P. Pu, F. Ricci, and M. Zanker, “Recommender systems: Trends and frontiers,” *AI Magazine*, vol. 43, no. 2, John Wiley & Sons, Ltd, pp. 145–150, 2022.
- [20] M. McTear, Z. Callejas, and D. Griol, *The conversational interface: Talking to smart devices*. Springer Cham, 2016.
- [21] J. Weizenbaum, “ELIZA - a computer program for the study of natural language communication between man and machine,” *Commun ACM*, vol. 9, no. 1, pp. 36–45, Jan. 1966.
- [22] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever, “Language Models are Unsupervised Multitask Learners,” *OpenAI blog*, vol. 1, p. 9, 2019.
- [23] A. Pal, A. Barigidad, and A. Mustafi, “IMDb Movie Reviews Dataset,” *IEEE Dataport*, 2020.
- [24] V. Sanh, L. Debut, J. Chaumond, and T. Wolf, “DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter,” *ArXiv*, p. 5, Oct. 2019.