ANÁLISIS DE SENTIMIENTOS EN LLAMADAS EN CENTROS DE ATENCIÓN AL CLIENTE

Andrea Arias Gómez

Daniel Alberto Rincón Loaiza

Jhon Alexander Rojas Tavera

Proyecto aplicado para optar al título de

Magíster en Ciencia de Datos

Director

Cristian Alejandro Torres V.

FACULTAD DE INGENIERÍA Y CIENCIAS

MAESTRÍA EN CIENCIA DE DATOS

SANTIAGO DE CALI, AGOSTO 03 DE 2025

TABLA DE CONTENIDO

LISTA DE FIGURAS

LISTA DE TABLAS

INTRODUCCIÓN

En el dinámico entorno empresarial de los centros de contacto, la calidad de las interacciones entre agentes telefónicos y clientes desempeña un papel esencial en la satisfacción del cliente y, por ende, en el crecimiento y reputación de las empresas. Sin embargo, la falta de herramientas para comprender y gestionar los sentimientos expresados durante estas interacciones compromete la capacidad de identificar áreas de mejora y optimizar la eficiencia operativa. En respuesta a esta problemática, surge la necesidad de implementar técnicas avanzadas de ciencia de datos para llevar a cabo un análisis de sentimiento en las llamadas en los centros de contacto.

Este proyecto se enfoca en el desarrollo de un modelo de clasificación de análisis de sentimientos en llamadas telefónicas, utilizando técnicas de aprendizaje automático. La iniciativa busca mejorar la comprensión de las interacciones agente-cliente, contribuyendo a los procesos de mejora continua y fortalecimiento de la calidad del servicio en los *Call Centers*. La relevancia de este proyecto radica en la necesidad de comprender las complejidades del lenguaje hablado en este contexto, con el objetivo de anticipar desafíos relacionados con la calidad del servicio y la gestión de experiencias del cliente. Los centros de contacto, generalmente respaldados por áreas de *Back Office* dedicadas al aseguramiento de la calidad, carecen de herramientas que faciliten la comprensión de los sentimientos expresados durante las interacciones de cada contacto. Esta carencia no solo compromete la identificación de experiencias insatisfactorias para los usuarios, sino que también puede resultar en la pérdida de clientes y en una percepción negativa de la marca.

El alcance del proyecto abarca la construcción de un corpus de datos representativo de las interacciones agente-cliente en centros de contacto en español. Se propone implementar y comparar diferentes algoritmos de clasificación, incluyendo modelos entrenados desde cero y un modelo preentrenado, con el fin de identificar y categorizar los sentimientos expresados durante las interacciones telefónicas. El proyecto también contempla la elaboración de una interfaz intuitiva que permita a los usuarios cargar nuevos registros para su análisis, asegurando la accesibilidad y usabilidad de la herramienta.

1. DEFINICIÓN DEL PROBLEMA

1.1. Planteamiento del problema

En el ámbito empresarial de los centros de contacto, la calidad resultante de las interacciones entre los agentes telefónicos y los clientes desempeña un papel esencial en la satisfacción de estos últimos y el éxito empresarial. Una experiencia negativa durante una interacción puede traducirse en pérdida de clientes, una percepción desfavorable de la marca, y un impacto negativo en los indicadores clave de desempeño empresarial. La capacidad de comprender lo que expresan los clientes durante estas llamadas y detectar sus sentimientos representa una oportunidad valiosa para mejorar la atención al cliente, optimizar la eficiencia operativa, y fomentar relaciones sostenibles entre las empresas y sus consumidores [1].

En este contexto, la ciencia de datos ofrece herramientas y enfoques avanzados que pueden transformar la forma en que se analizan las interacciones en los centros de contacto. En particular, el análisis de sentimientos aplicado a las transcripciones de las llamadas telefónicas permite extraer patrones emocionales y tendencias clave, que no solo proporcionan una visión más profunda de las percepciones y expectativas de los clientes, sino que también genera *insights* accionables para la toma de decisiones estratégicas. Este tipo de análisis facilita la detección temprana de áreas de mejora y permite la personalización de la experiencia del cliente en función de sus necesidades y emociones expresadas.

En Colombia, el sector del *Business Process Outsourcing (BPO*) representa una contribución significativa del 3,5 % del Producto Interno Bruto (PIB) nacional, con presencia en 26 de los 32 departamentos del país. Con el rápido crecimiento del sector, se ha incrementado la complejidad de los desafíos asociados con la gestión y el análisis de grandes volúmenes de datos generados por los centros de contacto. Sin embargo, la falta de herramientas avanzadas de análisis que integren técnicas de procesamiento de lenguaje natural (PLN) y aprendizaje automático compromete la capacidad de los centros de contacto para identificar de manera efectiva las experiencias insatisfactorias y los problemas recurrentes que afectan la calidad del servicio [2].

Tradicionalmente, los centros de contacto cuentan con áreas de *BackOffice* dedicadas al aseguramiento de la calidad, pero estas dependen en gran medida de revisiones manuales y subjetivas. Este enfoque limita la capacidad de las empresas para gestionar sentimientos expresados durante las interacciones, lo que resulta en una comprensión incompleta de las expectativas del cliente. La falta de un enfoque basado en datos no solo reduce la eficiencia operativa, sino que también restringe la capacidad de las organizaciones para anticipar tendencias y adaptarse a las demandas cambiantes del mercado [3].

Por lo tanto, es imperativo adoptar un enfoque sistemático basado en la ciencia de datos que permita analizar y gestionar de manera estructurada los sentimientos expresados por los clientes durante sus interacciones con los centros de contacto. La implementación de técnicas avanzadas de PLN, combinadas con modelos de aprendizaje automático, ofrece una oportunidad para automatizar y optimizar la detección de emociones, mejorar la experiencia del cliente y fortalecer la reputación de la empresa. Este enfoque no solo responde a los desafíos actuales de la industria, sino que también posiciona a las organizaciones para anticipar y resolver proactivamente problemas relacionados con la calidad del servicio, maximizando tanto la satisfacción del cliente como la eficiencia operativa.

1.2. Formulación del problema

En este segmento, se pretende contextualizar el desarrollo del proyecto aplicado mediante la formulación de la siguiente pregunta problema: ¿Cómo desarrollar un modelo de clasificación que emplee técnicas de aprendizaje automático para analizar los sentimientos del cliente a partir de los registros generados en los centros de contacto telefónico?

En este sentido, se derivan las siguientes preguntas de sistematización para abordar de manera estructurada la resolución del problema:

¿Cómo realizar un eficiente preprocesamiento de la información contenida en los registros de llamadas para optimizar la calidad de los datos utilizados en el modelado? ¿Cuáles son los métodos más adecuados para construir y optimizar modelos que permitan analizar los sentimientos del cliente? ¿Cómo evaluar la confiabilidad de los resultados obtenidos por el modelo, garantizando su aplicabilidad y relevancia en un entorno real? ¿Cuáles son los requerimientos de diseño que deben ser desarrollados para la construcción de una interfaz que permita al usuario final cargar nuevos registros para su respectivo análisis? ¿Cómo poner la herramienta a disposición de los posibles usuarios de forma óptima y accesible?

2. OBJETIVOS DEL PROYECTO

2.1. Objetivo general

Desarrollar un modelo de clasificación para el análisis de sentimientos en llamadas telefónicas de centros de atención al cliente utilizando técnicas avanzadas de aprendizaje automático.

2.2. Objetivos específicos

- Preprocesar la información contenida en los registros de llamadas, aplicando técnicas de reconocimiento automático del habla y procesamiento de lenguaje natural para optimizar la calidad del corpus utilizado en el modelado de análisis de sentimientos del usuario.
- Estructurar la base de datos con respecto al corpus obtenido para los registros de transcripción del usuario, mediante la asignación de etiquetas relacionadas con los sentimientos identificados.
- Entrenar algoritmos de clasificación que permitan la identificación, categorización y optimización de los sentimientos expresados durante las interacciones telefónicas, utilizando herramientas de aprendizaje profundo.
- Validar la precisión del modelo en un entorno controlado, a partir de las métricas específicas de evaluación de desempeño en tareas de análisis de sentimientos.
- Desarrollar los requerimientos de diseño necesarios para la construcción de una interfaz intuitiva que permita al usuario final cargar nuevos registros, facilitando así el análisis de sentimientos de manera eficiente con la incorporación de datos al corpus.

3. MARCO DE REFERENCIA

3.1. Marco Teórico

En esta sección se hace una visión de los conceptos clave que son la base teórica de nuestro proyecto. Estos conceptos son fundamentales para comprender la problemática abordada y la metodología propuesta.

3.1.1. Contexto

El centro de atención de llamadas (o *Call Center*) de la Pontificia Universidad Javeriana de Cali es un canal esencial para la comunicación entre la institución y sus potenciales estudiantes, actuales alumnos y padres de familia. Este canal gestiona un alto volumen de interacciones telefónicas, que incluyen desde consultas sobre programas académicos hasta solicitudes de apoyo financiero, procesos de inscripción y matrícula y reclamos.

Sin embargo, estas conversaciones, no se analizan sistemáticamente, lo que impide a la universidad entender a profundidad las percepciones, necesidades y preocupaciones de los interlocutores, especialmente de parte del usuario. Este vacío en el análisis representa una oportunidad perdida para mejorar la experiencia de los usuarios, optimizar procesos internos y fortalecer la imagen institucional. Además, puede llevar a que un estudiante potencial pierda interés en inscribirse o en continuar con el proceso de admisión, afectando así la misión educativa de la institución.

El análisis de sentimientos en este contexto se enfoca en identificar y clasificar las emociones expresadas por el usuario durante las llamadas al *Call Center*. Este enfoque permite detectar si el sentimiento general del usuario es positivo, negativo o neutro, brindando información clave para comprender su percepción de la interacción.

Relevancia en el contexto empresarial y educativo

- **Mejora en la experiencia del usuario:** Comprender los sentimientos expresados por los usuarios en las llamadas permite a la universidad anticipar problemas y resolver dudas de manera más empática y eficiente. Por ejemplo, un usuario que muestra frustración durante una consulta sobre becas puede estar indicando la necesidad de mejorar la claridad en la comunicación de los requisitos y procesos [4].
- **Retención de estudiantes potenciales:** Un análisis adecuado de las emociones puede ayudar a detectar interés genuino o indecisión en los prospectos que llaman

para conocer programas académicos. Esto permite a la universidad personalizar sus estrategias de seguimiento y aumentar las tasas de inscripción.

- **Optimización de recursos humanos y tecnológicos:** Los patrones emocionales pueden servir como indicadores para rediseñar procesos automatizados o incorporar herramientas de autoservicio más intuitivas [4].

3.1.2. Fundamentos teóricos y conceptuales

Esta sección establece las bases conceptuales necesarias para comprender el desarrollo del proyecto, enfocándose en los conceptos centrales de análisis de sentimientos, procesamiento de lenguaje natural (PLN) y aprendizaje automático.

3.1.2.1. Introducción al análisis de sentimientos

El análisis de sentimientos, también conocido como minería de opiniones, es una técnica del procesamiento de lenguaje natural (PLN) que permite identificar, extraer y clasificar las emociones y opiniones expresadas en un texto [4]. En esencia, esta técnica busca desentrañar el estado emocional subyacente en palabras o frases, proporcionando una visión estructurada y cuantificable de un fenómeno tan subjetivo como las emociones humanas. Los resultados del análisis se presentan típicamente como polaridades que se clasifican en tres categorías principales: positiva, negativa y neutra. Este enfoque ha ganado relevancia debido a su capacidad para procesar grandes volúmenes de datos no estructurados y transformarlos en información útil y accionable [5].

El propósito principal del análisis de sentimientos es comprender cómo las personas perciben productos, servicios, eventos o marcas, lo que resulta invaluable en contextos donde la satisfacción del usuario o cliente es un indicador clave del éxito. A diferencia de otros métodos de análisis textual, el análisis de sentimientos se centra específicamente en los aspectos emocionales y subjetivos del lenguaje, lo que lo convierte en una herramienta esencial para evaluar percepciones y opiniones en tiempo real [6].

En el contexto educativo, el análisis de sentimientos también encuentra aplicaciones significativas. Por ejemplo, las instituciones pueden analizar las interacciones entre estudiantes y personal administrativo para identificar patrones de satisfacción o frustración. Este enfoque permite no solo detectar problemas comunes, como la falta de claridad en los procesos administrativos, sino también implementar soluciones personalizadas para mejorar la experiencia del usuario. Asimismo, en centros de atención universitaria, el análisis de sentimientos aplicado a las llamadas de estudiantes puede revelar emociones negativas

asociadas con procesos burocráticos complicados, lo que proporciona una base sólida para rediseñar políticas y procedimientos.

3.1.2.2. Procesamiento de Lenguaje Natural (PLN)

El procesamiento de lenguaje natural (PLN) es una subdisciplina de la inteligencia artificial que se centra en la interacción entre las máquinas y el lenguaje humano. Su objetivo principal es permitir que los sistemas interpreten, analicen y generen texto o habla de manera que sea comprensible para los humanos y procesable por las máquinas. Para lograrlo, el PLN combina técnicas de lingüística computacional y aprendizaje automático, lo que lo convierte en una disciplina esencial para abordar problemas relacionados con el lenguaje [7].

Relación entre el PLN y el análisis de sentimientos

El PLN proporciona las herramientas necesarias para realizar el análisis de sentimientos, ya que permite a los sistemas comprender y procesar las complejidades del lenguaje natural. Las técnicas de preprocesamiento, como la ***tokenización*** (dividir el texto en palabras o frases), la **eliminación de *stopwords*** (palabras comunes que no aportan significado, como "el" o "y") y la **normalización del texto** (eliminar caracteres especiales o convertir todo el texto a minúsculas), son pasos iniciales imprescindibles para preparar los datos textuales.

La integración del PLN con el análisis de sentimientos es esencial, ya que muchas emociones y opiniones no se expresan de manera explícita, sino a través de contextos complejos, modismos o estructuras lingüísticas implícitas [6].

A pesar de sus avances, el PLN enfrenta desafíos importantes, especialmente en tareas relacionadas con la interpretación del lenguaje humano en análisis de sentimientos. La ambigüedad lingüística es uno de los problemas más persistentes. Muchas palabras tienen significados que dependen del contexto en el que se usan. Resolver estas ambigüedades requiere sistemas que no solo analicen las palabras individualmente, sino que también comprendan el significado de las oraciones completas y el documento en su conjunto [7].

Otro reto significativo es la detección de sarcasmo e ironía. Estas formas de expresión suelen invertir el significado literal de las palabras para transmitir un sentimiento opuesto. Por ejemplo, en la frase *"¡Qué servicio tan rápido!"*, un cliente insatisfecho podría estar expresando (con sarcasmo) frustración en lugar de admiración. Este tipo de complejidad emocional es difícil de detectar, ya que depende de señales contextuales sutiles y, a menudo, del conocimiento previo del lector o sistema. Los enfoques tradicionales basados en palabras

clave suelen fallar en este tipo de interpretaciones, lo que subraya la necesidad de métodos más avanzados, como modelos de aprendizaje profundo, que puedan captar estos matices [8].

3.1.2.3. Aprendizaje automático

El aprendizaje automático se define como el proceso que utiliza modelos matemáticos de datos para capacitar a un sistema a aprender sin instrucciones directas, emulando la capacidad de análisis y aprendizaje humano. En esencia, se trata de la ciencia de entrenar a los equipos para analizar y aprender a partir de los datos de manera análoga al proceso cognitivo humano. Este enfoque se encuentra dentro del ámbito de la inteligencia artificial (IA) y emplea algoritmos para identificar patrones en conjuntos de datos. Estos patrones se usan después para desarrollar un modelo de datos capaz de hacer predicciones. A medida que el sistema acumula más experiencia y datos, la precisión de las predicciones mejora, reflejando un paralelismo con la mejora humana a través de la práctica continua [9].

A diferencia de los sistemas tradicionales, donde el comportamiento del software es estrictamente definido por reglas programadas, el aprendizaje automático utiliza los datos como su principal fuente de conocimiento y evolución. El proceso del aprendizaje automático comienza con la recopilación y preparación de datos, que se utilizan para entrenar un modelo. Durante esta etapa, el modelo analiza los datos de entrada y ajusta sus parámetros internos para optimizar su capacidad de hacer predicciones precisas o tomar decisiones informadas. Una vez entrenado, el modelo se evalúa utilizando nuevos datos para medir su desempeño y capacidad de generalización. Este ciclo continuo de aprendizaje y evaluación es la base de su éxito y su capacidad para manejar tareas complejas como el análisis de sentimientos, el reconocimiento de imágenes y la predicción de tendencias [10].

Categorías principales del aprendizaje automático

El aprendizaje automático se divide en varias categorías según el tipo de datos disponibles y la naturaleza de la tarea a resolver. Entre las más relevantes se encuentran:

- **Entrenamiento supervisado:** En este enfoque, los modelos se entrenan utilizando datos etiquetados, donde cada entrada está asociada con una salida esperada. Por ejemplo, en el análisis de sentimientos, un modelo supervisado puede aprender a clasificar textos como positivos, negativos o neutros a partir de un conjunto de datos previamente etiquetado. Algoritmos como **Naive Bayes** y **Support Vector Machines (SVM)** son ejemplos populares de esta categoría. Estos modelos destacan por su simplicidad y eficiencia en tareas donde las etiquetas son claras y consistentes,

aunque su desempeño puede verse afectado en escenarios con datos ambiguos o altamente complejos [7].

- **Entrenamiento no supervisado:** Este enfoque trabaja con datos sin etiquetas, identificando patrones o estructuras ocultas en el conjunto de datos. Los algoritmos de *clustering*, como *K-means*, agrupan elementos similares basándose en sus características. Aunque menos común en el análisis de sentimientos directo, este enfoque puede ser útil para descubrir categorías emocionales o patrones inesperados en grandes volúmenes de texto no etiquetado [11].

3.1.3. Enfoques y métodos para el análisis de sentimientos

El proceso de análisis de sentimientos combina enfoques tradicionales basados en reglas con métodos modernos que emplean aprendizaje automático y aprendizaje profundo. A continuación, se presentan los principales enfoques y metodologías utilizados para abordar esta tarea.

3.1.3.1. Métodos basados en etiquetas y diccionarios

Los métodos basados en reglas y diccionarios representan el enfoque más tradicional del análisis de sentimientos. Estas técnicas utilizan listas predefinidas de palabras, conocidas como diccionarios de polaridad, que contienen términos asociados con emociones positivas, negativas o neutras. Además, emplean reglas gramaticales simples para analizar las relaciones entre palabras en una oración y determinar la polaridad general del texto.

VADER (Valence Aware Dictionary and Sentiment Reasoner)

Es una herramienta de análisis de sentimientos esencial para determinar la polaridad de un texto, es decir, si el contenido transmite emociones positivas, negativas o neutras. Su aplicación es valiosa en la evaluación de sentimientos en textos provenientes de redes sociales y comentarios en línea. Este enfoque utiliza un conjunto de reglas gramaticales y un diccionario predefinido de palabras con sus respectivas polaridades para asignar puntuaciones a las palabras y frases en el texto. Dichas puntuaciones se amalgaman para calcular la polaridad general del contenido, brindando una comprensión holística de la expresión emocional contenida en el texto [12]. VADER es especialmente útil en el análisis de textos breves y no estructurados, como publicaciones en redes sociales y comentarios en línea [13].

3.1.3.2. Métodos basados en Aprendizaje Automático Supervisado

Naive Bayes

Es un conjunto de algoritmos de clasificación supervisada basados en el teorema de Bayes, que asume una independencia condicional entre las características de los datos. A pesar de esta suposición simplista, Naive Bayes ha demostrado ser altamente efectivo en tareas como la clasificación de texto, especialmente en problemas como el análisis de sentimientos y la detección de spam [7]. Este modelo calcula la probabilidad de que una instancia pertenezca a una clase determinada, considerando la probabilidad conjunta de las características dadas.

El principal atractivo de Naive Bayes radica en su simplicidad y eficiencia computacional. Es capaz de manejar grandes conjuntos de datos con rapidez, lo que lo hace especialmente útil en aplicaciones donde el tiempo de procesamiento es crítico. Además, su rendimiento es robusto incluso en contextos de alta dimensionalidad, como los conjuntos de datos derivados de representaciones de texto como TF-IDF o *bag-of-words* [14].

Sin embargo, en escenarios donde las características están altamente correlacionadas, su precisión puede disminuir considerablemente. Por ello, su efectividad es mayor en dominios donde las variables realmente son independientes o cuando la correlación entre las características no es significativa [15].

Support Vector Machines (SVM)

Son técnicas robustas de aprendizaje automático utilizadas principalmente para tareas de clasificación y regresión. SVM se basa en encontrar un hiperplano óptimo que separe las clases en un espacio de características, maximizando el margen entre las muestras más cercanas de cada clase, conocidas como vectores de soporte [16]. Este enfoque lo hace especialmente adecuado para problemas de alta dimensionalidad, como la clasificación de texto, donde los datos poseen una gran cantidad de características.

Una característica destacada de SVM es su capacidad para manejar casos en los que las clases no son linealmente separables mediante el uso de funciones *kernel*. Estas funciones transforman los datos en un espacio de mayor dimensionalidad, permitiendo que las clases se separen linealmente en este nuevo espacio. Entre los *kernels* más utilizados están el *kernel* lineal, polinómico y radial (RBF), siendo este último particularmente efectivo para problemas complejos y no lineales [7].

En el análisis de texto, SVM ha demostrado ser muy eficiente en tareas como la clasificación de sentimientos, donde los datos textuales se convierten en representaciones numéricas mediante técnicas como TF-IDF o *bag-of-words*. SVM no solo logra buenos resultados en términos de precisión, sino que también es menos susceptible al sobreajuste en comparación con otros modelos, especialmente cuando se emplean técnicas de regularización [14].

A pesar de sus fortalezas, el entrenamiento de un modelo SVM puede ser computacionalmente costoso en conjuntos de datos muy grandes, ya que el tiempo de entrenamiento crece cuadráticamente con el número de muestras. Por esta razón, SVM se utiliza más comúnmente en proyectos con un tamaño de datos moderado o se combina con técnicas de reducción de dimensionalidad como PCA (Análisis de Componentes Principales) para mejorar su eficiencia [16].

Random Forest

Es un algoritmo de aprendizaje automático basado en árboles de decisión que opera como un conjunto (*ensemble learning*). Este modelo combina múltiples árboles de decisión independientes entrenados en subconjuntos diferentes de los datos y realiza un promedio (para regresión) o votación mayoritaria (para clasificación) para generar una predicción final. Este enfoque mejora significativamente la precisión y la robustez del modelo al reducir el riesgo de sobreajuste que podría ocurrir con un único árbol de decisión [17].

Random Forest se caracteriza por su capacidad para manejar conjuntos de datos con un gran número de características y observaciones, lo que lo hace particularmente adecuado para tareas de clasificación complejas como el análisis de sentimientos. El algoritmo utiliza la técnica de *bagging* (bootstrap aggregating) para entrenar cada árbol en una muestra aleatoria de los datos de entrenamiento. Además, en cada nodo, se selecciona un subconjunto aleatorio de características para decidir la división óptima, lo que introduce diversidad entre los árboles y mejora la generalización del modelo [18].

En el contexto del análisis de sentimientos, Random Forest se utiliza comúnmente como un modelo base para comparar su desempeño con algoritmos más complejos como *Gradient Boosting Machines* o redes neuronales, debido a su balance entre simplicidad, interpretabilidad y precisión.

MLP con TF-IDF

Los perceptrones multicapa (MLP) combinados con representaciones TF-IDF son una aproximación tradicional pero efectiva para el análisis de sentimientos. TF-IDF transforma el texto en vectores numéricos que reflejan la importancia de las palabras, y el MLP aprende patrones no lineales en estos vectores para clasificar el sentimiento. Aunque menos sofisticados que los modelos de aprendizaje profundo, ofrecen una buena base y son útiles en escenarios con recursos limitados [19].

3.1.3.3. Métodos basados en Aprendizaje Profundo

Es una rama del aprendizaje automático que utiliza arquitecturas neuronales compuestas por múltiples capas para modelar y extraer características complejas de los datos. Estas redes son particularmente efectivas en el procesamiento de lenguaje natural (PLN), ya que pueden aprender representaciones contextuales y jerárquicas del texto, lo que es esencial para el análisis de sentimientos [20].

En el contexto del análisis de sentimientos, las redes neuronales profundas se utilizan para clasificar texto en categorías de sentimientos, como positivo, negativo o neutro. A diferencia de los algoritmos tradicionales, las redes profundas pueden capturar relaciones complejas entre palabras, frases y contextos. Las principales arquitecturas utilizadas incluyen:

Transformers

Representan una arquitectura revolucionaria en el aprendizaje profundo, particularmente en el procesamiento de lenguaje natural (PLN). Esta arquitectura eliminó la necesidad de modelos secuenciales como las RNN y LSTM (*Long-Short Term Memory*) al implementar un mecanismo de atención que procesa las palabras de una secuencia en paralelo, asignando pesos contextuales a cada palabra con respecto a las demás en una oración [21].

BERT (Bidirectional Encoder Representations from Transformers): BERT es un modelo preentrenado que emplea una arquitectura de *Transformers* bidireccional. Esto significa que BERT considera el contexto tanto a la izquierda como a la derecha de una palabra dentro de una secuencia, a diferencia de modelos unidireccionales que solo procesan información en una dirección [22]. Su preentrenamiento incluye dos tareas principales:

- Máscara de palabras (*Masked Language Model - MLM*): Oculta ciertas palabras en una oración y entrena el modelo para predecirlas basándose en el contexto bidireccional.
- Predicción de la siguiente oración (*Next Sentence Prediction - NSP*): Evalúa la relación entre pares de oraciones, permitiendo captar relaciones contextuales a nivel de documento.

GPT (*Generative Pre-trained Transformer*): GPT es un modelo preentrenado enfocado en la generación de texto y modelado unidireccional, lo que significa que procesa el texto palabra por palabra, considerando únicamente el contexto hacia atrás. Sin embargo, su preentrenamiento masivo en corpus grandes lo hace extremadamente efectivo en tareas de generación y clasificación textual [23].

GPT utiliza datos no etiquetados para aprender patrones del lenguaje en tareas generales, como completar oraciones o responder preguntas. Una vez preentrenado, el modelo puede ajustarse para tareas específicas, como análisis de sentimientos, proporcionando una alta precisión en la predicción.

DistilBETO: Es una versión reducida y optimizada del modelo BERT entrenado específicamente para el idioma español. Al igual que DistilBERT, mantiene un rendimiento comparable al modelo original, pero con menor tamaño y mayor eficiencia computacional, lo que lo hace adecuado para aplicaciones con recursos limitados. Este modelo ha demostrado ser efectivo en tareas de análisis de sentimientos en español, permitiendo una comprensión profunda del contexto lingüístico [24].

Otras redes neuronales

Redes Neuronales Recurrentes (RNN): Diseñadas para manejar datos secuenciales, como texto. Capturan dependencias a lo largo de una secuencia, lo que las hace útiles para modelar el flujo lógico en las oraciones [22].

Redes Neuronales Convolucionales (CNN): Son un tipo específico de arquitectura dentro del aprendizaje profundo diseñada para trabajar especialmente bien con datos en forma de cuadrículas, como imágenes. Las CNN utilizan capas de convolución para aprender patrones locales en los datos [12, p.359].

TextCNN: Es una arquitectura de red neuronal convolucional diseñada para tareas de clasificación de texto, incluyendo el análisis de sentimientos. Utiliza filtros convolucionales para capturar características locales y patrones *n-gram* en el texto, seguido de capas de

pooling y clasificación. Esta arquitectura ha demostrado ser efectiva en la clasificación de sentimientos en textos cortos y no estructurados, como publicaciones en redes sociales.

Bi-LSTM: Las redes LSTM bidireccionales (Bi-LSTM) son una extensión de las LSTM tradicionales que procesan la información en ambas direcciones, capturando contextos pasados y futuros en una secuencia de texto. Esta capacidad las hace especialmente útiles para el análisis de sentimientos, donde el contexto completo de una oración puede influir en la interpretación emocional [25].

Bi-GRU: Las redes GRU bidireccionales (Bi-GRU) son similares a las Bi-LSTM pero con una arquitectura más simple y menos parámetros, lo que las hace más eficientes computacionalmente. Han demostrado un rendimiento comparable en tareas de análisis de sentimientos, especialmente en conjuntos de datos grandes y variados [26].

Modelo Híbrido CNN-LSTM: Los modelos híbridos que combinan redes convolucionales (CNN) y redes de memoria a largo plazo (LSTM) aprovechan las fortalezas de ambas arquitecturas: las CNN capturan características locales y patrones en el texto, mientras que las LSTM modelan dependencias a largo plazo. Esta combinación ha demostrado mejorar la precisión en tareas de análisis de sentimientos al capturar tanto características locales como contextuales [27].

Método Semiautomático

En el ámbito del aprendizaje automático se refiere a un enfoque en el cual el modelo trabaja en colaboración con un humano durante determinadas etapas del proceso. Este enfoque integra la capacidad analítica de las máquinas con la experiencia y juicio humano. En el contexto de clasificación de nuestros datos, el proceso de etiquetado semiautomático implica que el modelo sea entrenado con ajustes identificados por los integrantes del proyecto en casos de clasificaciones incorrectas o ambiguas. Este proceso iterativo de retroalimentación entre la máquina y el humano se repite hasta lograr un nivel satisfactorio de precisión y calidad en las predicciones [28].

3.1.3.3.1. Consideraciones prácticas en el entrenamiento de modelos de aprendizaje profundo

El entrenamiento de modelos de aprendizaje profundo para el análisis de sentimientos requiere tener en cuenta diversas técnicas y decisiones prácticas que inciden directamente en la capacidad predictiva y generalización del sistema. Estas consideraciones no solo complementan la elección de arquitectura, sino que determinan la eficacia con la que el

modelo enfrenta los retos inherentes al procesamiento del lenguaje natural en contextos reales, como el desbalance de clases, la variabilidad del lenguaje o las limitaciones computacionales.

A continuación, se describen algunas de las prácticas más relevantes aplicadas durante el entrenamiento de los modelos utilizados en este proyecto.

a) **Desbalance de clases y estrategias de balanceo:** En muchas tareas de clasificación de sentimientos, especialmente en entornos reales como los centros de llamadas, se presenta un desbalance de clases significativo. Por ejemplo, la clase "Neutral" puede representar más del 60 % del corpus, lo que puede llevar a que el modelo favorezca predicciones de esa clase, reduciendo su sensibilidad hacia las clases minoritarias ("Positivo" y "Negativo"). Para mitigar este fenómeno, se emplean técnicas como:

   - **Oversampling:** replicar ejemplos de las clases minoritarias para equilibrar la distribución del conjunto de entrenamiento.
   - **Undersampling:** reducir aleatoriamente la cantidad de ejemplos de la clase mayoritaria.
   - **Función de pérdida ponderada (*weighted loss*):** ajustar la función de error para penalizar más los errores cometidos en las clases con menor representación.

   Estas estrategias permiten al modelo aprender patrones relevantes en todas las clases, optimizando métricas como el F1-score macro y garantizando una clasificación más equitativa.

b) **Tokenización en modelos Transformer:** Los modelos basados en la arquitectura Transformer, como BERT o DistilBETO, utilizan *tokenizadores* avanzados como WordPiece o SentencePiece, que fragmentan las palabras en subunidades léxicas denominadas *subwords*. Este enfoque resulta especialmente útil para manejar vocabularios extensos y adaptarse a palabras desconocidas o poco frecuentes [22].

   Además, en BERT se emplean tokens especiales, como [CLS], que se añade al inicio de cada secuencia y cuya representación vectorial final es utilizada por el modelo para tareas de clasificación. Esta tokenización específica es un componente esencial del proceso de entrada de datos en estos modelos, y difiere significativamente de los métodos tradicionales de segmentación por palabras.

c) **Ajuste fino (*fine-tuning*) de modelos preentrenados:** Modelos como BETO o DistilBETO han sido preentrenados con grandes volúmenes de texto en español mediante tareas genéricas como el *enmascaramiento de palabras* (*masked language modeling*).

Para adaptarlos a tareas específicas, como la clasificación de sentimientos en llamadas telefónicas, se aplica una técnica denominada **fine-tuning** [22].

El *fine-tuning* consiste en continuar el entrenamiento del modelo preentrenado, pero sobre un corpus etiquetado con una tarea concreta, ajustando todos sus pesos internos. Esta técnica permite aprovechar el conocimiento lingüístico general aprendido durante el preentrenamiento, a la vez que se especializa el modelo en la tarea deseada [22].

d) Entornos colaborativos de entrenamiento: La implementación y entrenamiento de modelos complejos de PLN se ha facilitado enormemente gracias a plataformas como **Google Colab**, que ofrecen acceso gratuito a recursos de cómputo acelerado (GPU y TPU), integración con bibliotecas como *transformers*, y soporte para entornos reproducibles y colaborativos.

Estos entornos permiten ejecutar experimentos avanzados, entrenar modelos con múltiples configuraciones y compartir resultados de forma ágil, sin requerir infraestructura local especializada. Su uso se ha convertido en una práctica estándar en proyectos académicos y experimentales de PLN.

En la fase posterior al entrenamiento, estos modelos pueden ser utilizados para realizar inferencias individuales sobre nuevos datos, lo cual permite su integración en aplicaciones prácticas para el análisis automático de sentimientos. Este proceso de despliegue se ve facilitado actualmente por el uso de *frameworks* de alto nivel como *HuggingFace*, que proporcionan estructuras modulares y reproducibles para el entrenamiento, evaluación y utilización de modelos de lenguaje natural. La disponibilidad de estas herramientas ha favorecido la adopción de modelos complejos incluso en contextos académicos o con recursos computacionales limitados, al permitir una implementación eficiente y accesible.

3.1.4. Técnicas del PLN aplicadas al análisis de sentimientos

- **Preprocesamiento:** El preprocesamiento de datos es una fase crítica en la preparación del texto para el análisis de sentimientos en el habla de centros de atención al cliente. Este proceso incluye técnicas clave diseñadas específicamente para adaptar el texto a las necesidades de este proyecto [12, p.174].
- **Eliminación de palabras:** La eliminación de palabras vacías e irrelevantes es un paso crucial en el preprocesamiento. Se identifican y eliminan aquellas palabras que no contribuyen significativamente al análisis de sentimientos, como artículos y preposiciones, permitiendo así una concentración más efectiva en las expresiones emocionales relevantes presentes en las interacciones telefónicas [12, p.405].

- **Tokenización:** Es el paso inicial en el preprocesamiento de texto, en este se divide el texto en unidades más pequeñas llamadas tokens. Estos tokens pueden ser palabras, frases o incluso caracteres, dependiendo de la granularidad deseada. La *tokenización* permite convertir cadenas de texto en una secuencia de unidades más manejables, facilitando así el análisis y la extracción de características. Además, facilita la representación numérica del texto, ya que cada token puede asignarse a un identificador único o vector de incrustación [12, p. 262 –263].
- **Vectores de incrustación:** Es una representación numérica de palabras que captura su significado y relaciones semánticas en un espacio multidimensional. Estos vectores son esenciales para que los modelos de lenguaje comprendan y procesen el texto de manera más efectiva. Palabras con significados similares tendrán vectores de incrustación cercanos entre sí. Por ejemplo, en un buen modelo de *embeddings*, las palabras "*puppy*" y "*dog*" estarán más cercanas que "*cat*" y "*houses*" Figura 1 [29].

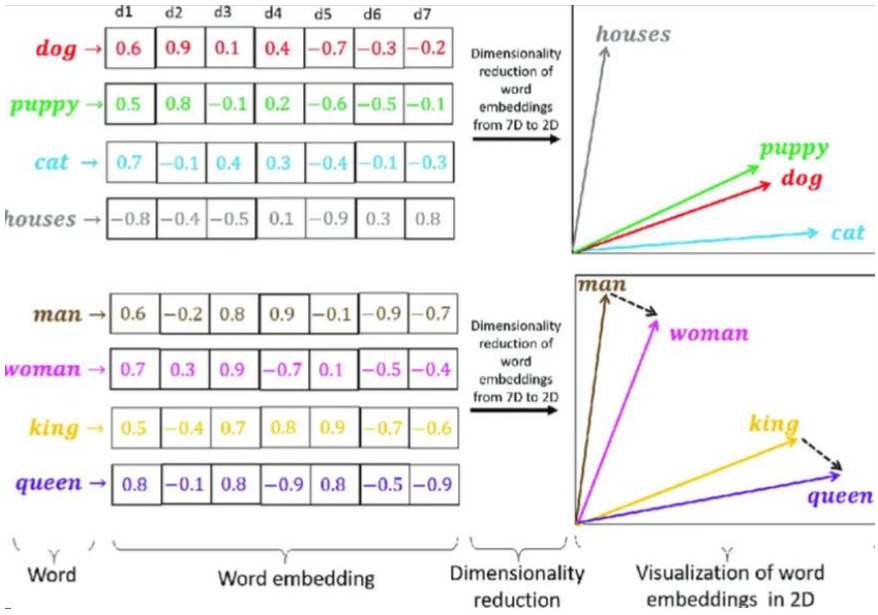


Figura 1. Ejemplos de incrustaciones de palabras y su respectiva representación gráfica.

- **Normalización:** La normalización transforma palabras u oraciones a su forma "canónica". En el contexto del análisis de sentimientos en centros de contacto, garantiza que las expresiones y términos relevantes se representen de manera consistente, independientemente de las variaciones en la forma de expresión [12, p.50].
- **Análisis de Sentimiento a Nivel de Entidad (ELSA):** Es una tarea específica dentro del procesamiento de lenguaje natural (NLP) que se centra en determinar la polaridad del sentimiento asociado con entidades específicas en un texto. En lugar de analizar el sentimiento general del texto completo, la atención se dirige a las entidades

individuales presentes en el texto, como nombres de productos, empresas, personas, lugares, etc. [4].

3.1.5. Métricas de evaluación

Para evaluar el rendimiento del sistema de ciencia de datos que se construye y verifica qué tan cerca se está del objetivo planteado, se necesitan utilizar funciones que puntúen el resultado. Normalmente, se utilizan diferentes funciones de puntuación para abordar problemas de clasificación binaria, clasificación multietiqueta, regresión o problemas de agrupamiento. A continuación, se describen las funciones métricas más populares para los algoritmos de clasificación multietiqueta que serán de importancia en nuestro desarrollo de análisis de sentimiento [30].

- **Precisión**: La precisión mide la proporción de instancias positivas correctamente identificadas entre todas las instancias clasificadas como positivas. En otras palabras, indica cuántas de las predicciones positivas del modelo son realmente correctas. Se calcula como $\frac{TP}{TP+FP}$, donde TP es el número de verdaderos positivos y FP es el número de falsos positivos [12, p.174].
- **Recall:** Evalúa la capacidad del modelo para identificar correctamente todas las instancias positivas. Es la proporción de verdaderos positivos entre la suma de verdaderos positivos y falsos negativos. Se calcula como $\frac{TP}{TP+FN}$, donde TP es el número de verdaderos positivos y FN es el número de falsos negativos [12, p.174].
- **F1-Score:** mide la proporción de predicciones correctas en el conjunto de datos. Es la suma de verdaderos positivos y verdaderos negativos dividida por el total de instancias. Su fórmula es $\frac{TP+TN}{TP+TN+FP+FN}$, donde TN es el número de verdaderos negativos [12, p.174].

3.2. Antecedentes

La interacción en los centros de atención telefónica es fundamental, pero entender el sentimiento expresado en estas conversaciones puede ser un desafío. Por una parte, primero se deben realizar las transcripciones de audio a texto, luego se debe crear una base de datos con cantidades similares de llamadas con sentimientos, sin tener una muestra de datos sesgada y, por último, entrenar un modelo que permita predicción de cada llamada para dar un valor agregado a estos datos.

En la literatura se tienen varios casos de estudio, en uno se encontraron con el desafío de construir una base de datos sin eventos de sentimiento a nivel de entidad, lo que podría

resultar en un conjunto de datos desequilibrado. Para superar esto, se empleó un modelo de reconocimiento de entidades (NER) basado en DistilBERT y una red neuronal convolucional (CNN) para muestrear 13.000 expresiones que contenían al menos una entidad nombrada con sentimiento positivo o negativo. Se agregaron 10.000 expresiones adicionales con al menos una entidad, pero sin sentimientos polarizados para equilibrar el conjunto de datos. Después, evaluadores independientes anotaron manualmente las 23.000 expresiones resultantes, que determinaron el sentimiento hacia la entidad e identificaron los términos de opinión. Este enfoque integral aseguró la construcción de un conjunto de datos equilibrado y anotado, capturando de manera efectiva los eventos de sentimiento a nivel de entidad en el contexto específico de las transcripciones telefónicas [4].

I work at Google and I love it a lot.
She's very impressed how MAC works so well.
He has hard time finding a good yogurt from Walmart.
It's quite difficult to navigate the mobile app of Instacart.

Figura 2. Ejemplos de análisis de sentimientos a nivel de entidad.

Como se puede observar en la Figura 2 una vez obtenido el texto existen diversos enfoques con los que podemos determinar el sentimiento asociado a entidades específicas en un texto. En lugar de analizar el sentimiento general del texto, como se hace en el análisis de sentimientos convencional, ELSA se centra en identificar y evaluar las emociones expresadas hacia entidades nombradas específicas. En el contexto de procesamiento de lenguaje natural, una entidad nombrada se refiere a cualquier objeto, lugar, persona o cosa con un nombre específico, como una marca, una persona famosa o una empresa. Siendo su objetivo capturar de manera precisa y automática el sentimiento asociado a entidades específicas en el texto, lo que es crucial para comprender las actitudes y percepciones hacia entidades particulares [31].

Tal como lo mencionan los autores del anterior artículo *Entity-level Sentiment Analysis in Contact Center Telephone Conversations* [4], uno de los principales desafíos de su modelo es que no logró identificar aquellas entidades por falta de datos, en el caso del proyecto que se está abordando, se tiene la ventaja de que, en caso de que se quisiera implementar un modelo similar al que ellos implementaron, no se tendría esta dificultad, debido a que todos los clientes se estarían refiriendo al servicio ofrecido por la PUJ, o el departamento de la universidad con la que se están comunicando. Por otra parte, resulta interesante ver la forma en la que los autores de este trabajo pudieron crear y anotar un conjunto de datos propio, dado que no contaban con un conjunto de datos públicamente disponible para la tarea de *Entity-level Sentiment Analysis* (ELSA), esto es un paradigma que

en este presente proyecto se podría llevar a cabo, basados en que inicialmente la PUJ otorgará un total de 1.000 muestras para el modelo, lo cual resulta ser una muestra muy pequeña.

Otra forma de realizar el análisis es segmentando la conversación por oraciones, identificando los sentimientos involucrados en cada oración de la conversación [32, Figura 3].

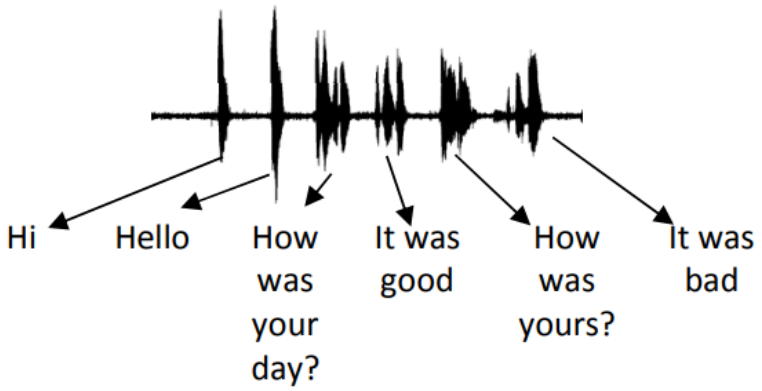


Figura 3. Segmentación de un audio entre dos personas por partes.

De esta forma se realiza un análisis de sentimiento para cada segmento del audio, como se ilustra en la Figura 4 [29- N33]. Esta imagen se extrajo de un estudio en el cual los responsables aplicaron este análisis utilizando la API de AssemblyAI, logrando la transcripción en tiempo real, así como la recopilación de sus métricas. Posteriormente, asignaron un sentimiento correspondiente a cada oración, definieron el inicio y el final de la conversación, y finalizaron agrupando todas las métricas de dicha interacción para determinar la naturaleza de la conversación entre las partes involucradas.

Sentence	Text	Duratio	Speake	Sentimer
1	Hi, Peter.	0.62	B	NEUTRAL
2	Did you hear they're going to make two graduation ceremonies this year?	3.57	B	NEUTRAL
3	Yeah, they say it's due to enrollment numbers.	2.19	A	NEUTRAL
4	Too many students getting their diplomas at once.	2.35	A	NEUTRAL
5	Make the ceremony last for hours.	1.98	A	NEUTRAL
6	I really don't like that.	1.92	B	NEGATIVE
7	Some of my friends are assigned to a different ceremony than mine.	3.03	B	NEUTRAL
8	We've been waiting for this day for years.	2.11	B	NEUTRAL
9	And now what?	1.16	B	NEUTRAL
10	Well, I mean, have you ever sat through the graduation ceremony?	3.91	A	NEUTRAL
11	I went last year and it lasted almost 3 hours.	3.36	A	NEUTRAL
12	And it took almost an hour to just read the names.	2.91	A	NEGATIVE
13	That's my point.	1.05	B	NEUTRAL
14	Almost 2 hours of the ceremony consists of speeches.	3.69	B	NEUTRAL
15	That's the real boring part.	1.71	B	NEGATIVE
16	Why don't they limit the number of speakers?	2.03	B	NEUTRAL
17	They could reduce the time of the ceremony over an hour if they just focused on the important people.	5.23	B	NEUTRAL
18	Yeah, I guess you're right.	1.77	A	NEUTRAL
19	But what about the auditorium?	1.68	A	NEUTRAL
20	Each student is only allowed two invitations to the ceremony.	3.18	A	NEUTRAL
21	But if they separate the events, then you can bring more of your family and friends.	3.86	A	NEUTRAL
22	Don't you want more people to come you.	1.95	A	NEUTRAL

Figura 4. Análisis de sentimiento por sentencia en una conversación.

En relación con el trabajo *Sentiment Analysis on Speaker Specific Speech Data* [32], es importante señalar que, a diferencia de ellos, el presente proyecto no empleará ninguna API. Si se quisieran transcribir los audios a texto se debería desarrollar un algoritmo para realizar esta conversión y construir una base de datos de audios con base en los campos que se observan en la Figura 4 [29]. Para definir si el análisis de sentimiento de una llamada se clasificará como positivo, negativo o neutral, el punto de referencia sería qué sentimiento predomina durante la llamada. También podría contemplarse otorgar un peso mayor a los sentimientos negativos o positivos, como se evidencia en el recuadro, la mayoría de las emociones son neutrales durante el desarrollo de la conversación; sin embargo, existen trazos de la conversación que pueden ser negativos y, aunque estos sean menos frecuentes que los neutrales, podrían ser un claro indicio de una conversación con sentimiento negativo. Cabe anotar que los autores de este articulo manifiestan el problema que encontraron cuando en diversos audios dos personas hablaban al tiempo, esto presenta un desafío para el modelo que perdería notoriamente su precisión y es un problema que se enfrentará en el desarrollo de este proyecto. Por último, se encontró que en este artículo utilizaron diferentes algoritmos para el análisis de sentimiento, como Naive Bayes, Linear SVM y VADER, los cuales se aplicarán en el desarrollo del presente proyecto.

4. PROPUESTA DE UN SISTEMA DE ANÁLISIS DE SENTIMIENTOS PARA TRANSCRIPCIONES DE AUDIOS EN CENTROS DE LLAMADAS DE ATENCIÓN AL CLIENTE

Este capítulo aborda el desarrollo integral de este proyecto, estructurado bajo un enfoque metodológico dual que combina las fortalezas de CRISP-DM y la metodología ágil Scrum. Este enfoque fue seleccionado para garantizar un proceso riguroso en el análisis de datos y la flexibilidad necesaria para gestionar un equipo distribuido geográficamente. A través de esta combinación, se buscó no solo cumplir con los objetivos planteados, sino también adaptarse a los desafíos inherentes al entorno del proyecto.

La metodología CRISP-DM (*Cross Industry Standard Process for Data Mining*) proporcionó una base estructurada para abordar las diferentes etapas del proyecto. Estas fases incluyen la comprensión del negocio, comprensión de los datos, preparación de los datos, modelado, evaluación y despliegue [34]. Este marco permitió al equipo avanzar de manera secuencial y lógica desde la identificación del problema hasta la implementación efectiva de una solución. Cada fase se alineó con los objetivos específicos de la investigación, facilitando un análisis exhaustivo y fundamentado.

Por otro lado, para asegurar un seguimiento ágil y eficiente del progreso del proyecto, se adoptó la metodología Scrum, que permitió realizar ciclos iterativos cortos y ajustes constantes durante el desarrollo del proyecto [35]. La combinación de ambas metodologías se alinea con prácticas documentadas en estudios recientes que integran enfoques ágiles con procesos de análisis de datos para maximizar la adaptabilidad y precisión [36]. Se diseñaron sprints que coincidieron estratégicamente con las etapas de CRISP-DM, permitiendo ciclos de iteración cortos y frecuentes. Esto posibilitó la revisión continua de los avances, la identificación de áreas de mejora y la realización de ajustes necesarios. En este contexto, se presenta a continuación la primera fase de CRISP-DM Comprensión del Negocio, en este contexto se identificó la problemática y los objetivos del proyecto desde una perspectiva aplicada. Este análisis inicial resultó fundamental para establecer un marco que guiara las decisiones técnicas y asegurara la alineación con los objetivos específicos definidos.

El problema identificado radicó en la dificultad de la Universidad Javeriana Cali para analizar eficientemente grandes volúmenes de interacciones telefónicas con clientes, un recurso valioso, pero poco explotado debido a su naturaleza no estructurada. Estas interacciones contienen información relevante para la mejora de los servicios, pero su análisis manual resulta costoso, lento y sujeto a subjetividades. En este contexto, el proyecto se propuso diseñar un sistema que no solo automatizara el procesamiento de estas interacciones, sino que también facilitara la extracción de *insights* relevantes mediante el análisis de los sentimientos expresados durante dichas conversaciones.

Durante esta fase, se llevaron a cabo diversas actividades que permitieron definir los fundamentos del proyecto:

a) **Calibración de audios:** Se llevaron a cabo sesiones de escucha de llamadas por parte del equipo de trabajo, con el objetivo de analizar patrones recurrentes, aspectos técnicos relevantes de las grabaciones y posibles desafíos relacionados con la calidad del audio. Durante estas sesiones, se identificó que la mayoría de los registros corresponden a gestiones ***outbound*** (llamadas salientes desde el *Call Center* con fines comerciales), mientras que el resto son ***inbound*** (llamadas entrantes dirigidas a brindar servicio al cliente). En estas llamadas se registran solicitudes como:

   - Consultas administrativas: Procesos de matrícula, becas, inscripciones.
   - Solicitudes de información: Programas académicos, horarios, costos.
   - Reclamos y soporte: Problemas técnicos o administrativos.

Además, se detectaron retos significativos para la transcripción de las llamadas, ya que en varios registros se percibieron tonos de voz muy bajos o la presencia de un alto nivel de ruido ambiental, lo que podría dificultar el proceso de interpretación y conversión de los audios en texto.

b) **Reuniones con los directores del proyecto**: Los directores, expertos en ciencia de datos, proporcionaron orientaciones clave para la definición del plan de trabajo. Estas reuniones ayudaron a identificar requisitos técnicos, como el flujo de datos, las limitaciones tecnológicas y las complejidades específicas del desarrollo.

A continuación, en la Figura 5, se presenta un diagrama de actividades que ilustra las principales etapas del proceso, desde el manejo de archivos de audio hasta la clasificación de sentimientos.

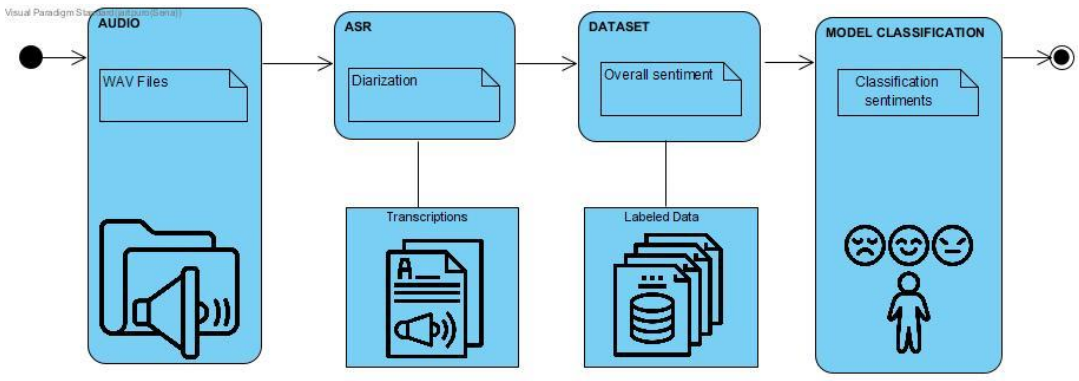


Figura 5. Etapas del proceso de análisis de sentimientos

A partir de las actividades realizadas, se identificaron diversas necesidades clave, lo que permitió establecer los principales requisitos del proyecto:

- **Integración de un sistema de reconocimiento automático del habla (ASR):** Implementación de tecnologías de alta precisión para transcribir las interacciones telefónicas de manera confiable.

- **Diarización de hablantes:** Uso de técnicas avanzadas que permitan identificar y separar las voces de los diferentes interlocutores en las grabaciones, facilitando un análisis individualizado más detallado.

- **Procesamiento de Lenguaje Natural (PLN):** Aplicación de métodos para normalizar y estructurar los datos textuales extraídos de las transcripciones, asegurando su calidad y utilidad en las etapas posteriores del análisis.

- **Análisis de sentimientos:** Desarrollo de algoritmos capaces de identificar y clasificar automáticamente las emociones expresadas durante las interacciones.

- **Interfaz escalable:** Diseño de una herramienta intuitiva y eficiente, construida con tecnologías de bajo consumo de recursos, que ofrezca versatilidad y facilidad de uso para los usuarios finales.

El diagrama refleja cómo estos requisitos se integran en un flujo lógico y estructurado. Este proceso comienza con la carga de archivos de audio en formato WAV, continúa con la transcripción y *diarización* mediante el ASR, y concluye con la clasificación final de sentimientos a través de algoritmos especializados.

5. PREPROCESAMIENTO DE LOS REGISTROS DE LLAMADAS

En este apartado se detalla la segunda fase de CRISP-DM, denominada *Comprensión de los datos*, cuyo objetivo principal es recopilar y analizar los datos disponibles, explorando tanto su estructura como su calidad inicial. Esta etapa es fundamental para garantizar que los datos utilizados en las fases posteriores sean confiables y relevantes [37]. En este contexto, se desarrolla el primero y segundo objetivo específico de este proyecto aplicado.

5.1. Exploración de los archivos

Se recibieron un total de 1.157 archivos de audio en formato WAV, configurados en estéreo, a través de enlaces temporales proporcionados por la plataforma WeTransfer. Estos archivos fueron entregados mediante el correo electrónico institucional, lo que garantizó un canal de transferencia de datos seguro y controlado. Este enfoque se utilizó para cumplir con las políticas de confidencialidad y privacidad establecidas por la institución, minimizando riesgos asociados a la exposición o acceso no autorizado a los datos.

Los archivos de audio fueron organizados en 3 entregas diferentes, se recibieron 260 archivos en la primera entrega identificada como "Grabaciones-1", 512 archivos en la segunda entrega denominada "Grabaciones-2" y 385 archivos en la tercera entrega identificada como "Grabaciones-3".

Para facilitar la transferencia de datos de forma temporal y evitar un almacenamiento permanente en servidores externos, los enlaces proporcionados por **WeTransfer** se configuraron con fechas de vencimiento predefinidas. Los enlaces utilizados fueron:

- Partes 1 y 2: https://wetransfer.com/downloads/4b64b5fed4ae8869bec92b6486530e6b20240508225003/7fb8ffa5eec9f461bd138b6fd9eed94720240508225003/043632
- Parte 3: https://wetransfer.com/downloads/c4e6a30b5920c2cdbb6f6197648e9d17202405090033328/51541154eda11232aff70e64ab7f1de820240509033328/54134a.

Una vez descargados, los archivos fueron almacenados y organizados de acuerdo con las buenas prácticas de gestión de datos, siguiendo estrictos protocolos de confidencialidad y asegurando su uso exclusivo para los fines definidos en este proyecto. Este proceso no solo

garantizó la protección de los datos sensibles, sino que también sentó las bases para su posterior análisis y preprocesamiento.

Este proceso permitió identificar que las llamadas presentaban una duración variable, con una media de 2,05 minutos. No obstante, se detectó la presencia de datos atípicos, ya que el rango de duración de las llamadas oscila entre 0,15 y 18,78 minutos [ver Tabla 1].

Tabla 1. Distribución cantidad de audios por minuto

Minutos	Cantidad
00	308
01	404
02	171
03	103
04	64
05	40
06	21
07	22
08	6
09	3
10	4
11	3
12	1
13	1
14	0
15	1
16	1
17	1
18	3
Total general	**1.157**

Se llevó a cabo una escucha completa de los 1.157 audios, lo que permitió identificar que la mayoría correspondían a gestiones de tipo *outbound*. Este hallazgo resultó crucial para definir el enfoque del proyecto, especialmente en la configuración del pipeline y el paradigma empleado en las etapas subsecuentes. A partir de este análisis, se identificó un patrón consistente en la estructura de las conversaciones, descrito de la siguiente manera:

- Speaker 0: Representa al agente del centro de atención al cliente.
- Speaker 1: Corresponde al usuario o cliente que recibe información académica o el ofrecimiento de algún programa.

Se realizó una exploración detallada de los archivos de audio utilizando herramientas especializadas para el tratamiento y análisis de audios, como *Adobe Audition*. Dado el contexto de nuestra investigación y los antecedentes establecidos, se determinó que la segmentación de los oradores podía lograrse mediante el análisis de las formas de onda y frecuencias espectrales. Este enfoque permitió avanzar significativamente en la definición de los planes de trabajo orientados a la transcripción de cada archivo.

En la Figura 6, se puede observar que una estrategia viable para obtener transcripciones con diarización consiste en aplicar técnicas específicas a cada canal de audio. Esto se debe a que la voz del cliente (Speaker 1) se encuentra predominantemente en el canal izquierdo, mientras que la voz del asesor (Speaker 0) está en el canal derecho. Este hallazgo facilita la implementación de un proceso más eficiente, centrado en desarrollar soluciones que aprovechen la separación de canales para el análisis y procesamiento de los datos.

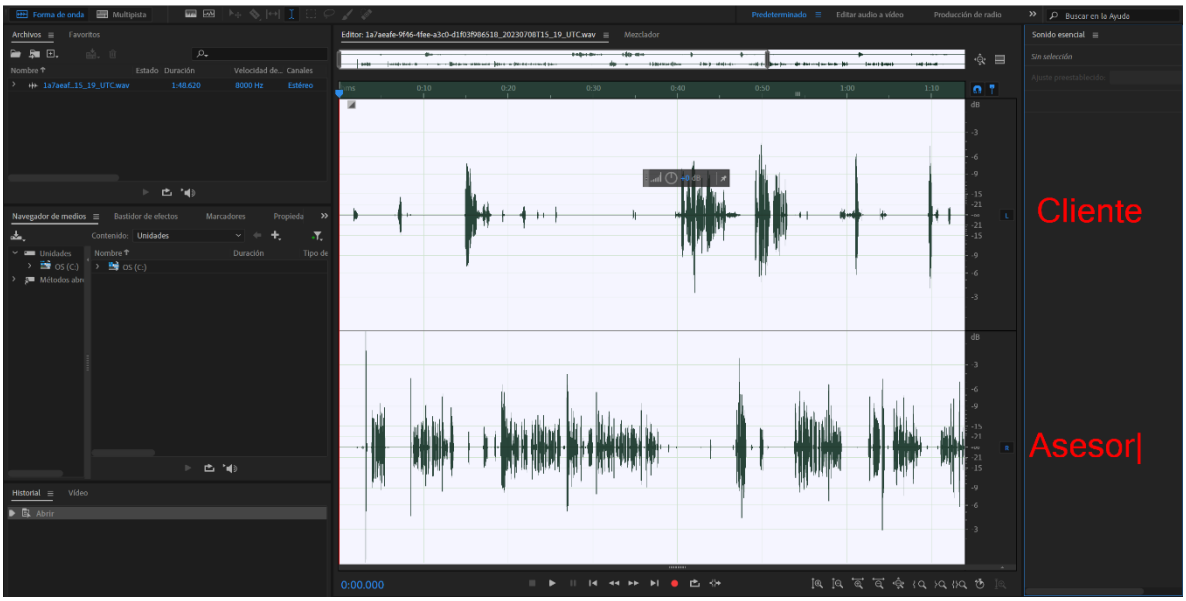


Figura 6. Formato de onda con voces separadas por canal en formato estéreo

En la Figura 7 muestra la frecuencia espectral de la señal de un archivo de audio, lo que permite explorar posibilidades de transcripción con diarización. Este enfoque no se limita a un canal específico dentro de la configuración estéreo, sino que aborda el espectrograma en su totalidad, identificando en la señal a los distintos hablantes presentes en la conversación.

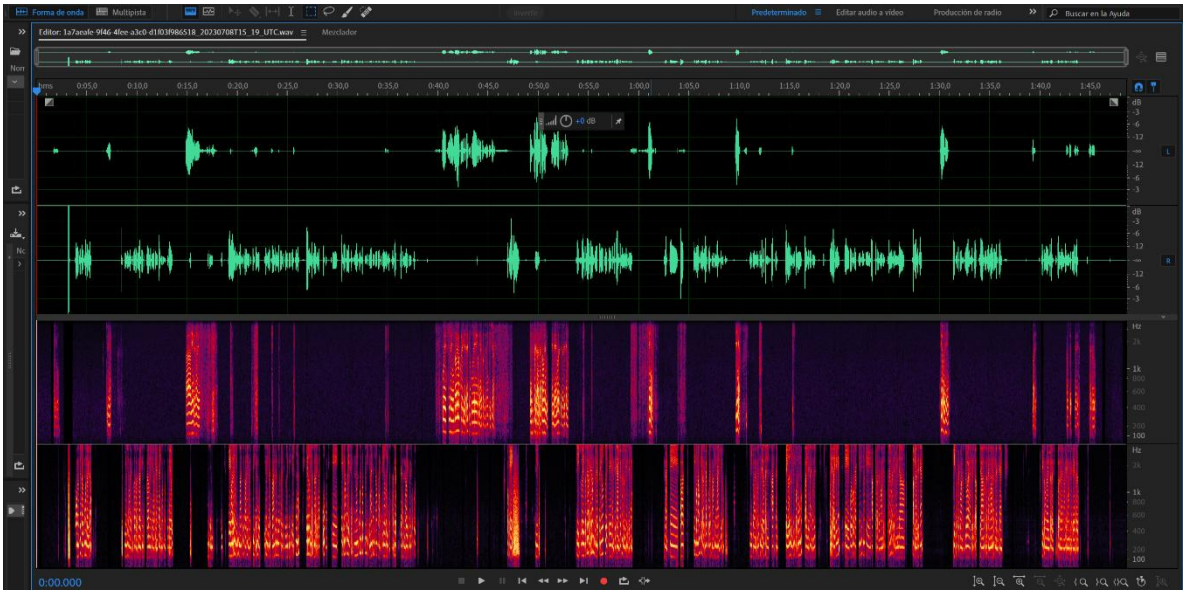


Figura 7. Formato de onda y frecuencia espectral de uno de los audios

De esta manera, se desarrolló un script que, debido a los altos requerimientos computacionales necesarios para su ejecución, fue implementado en Google Colab. Para optimizar el rendimiento, se utilizaron diferentes sesiones de máquinas virtuales, las cuales debían cumplir con requisitos específicos, como contar con tarjetas gráficas T4 y soporte para procesamiento CUDA en GPU. Esta configuración permitió llevar a cabo las tareas de manera eficiente, aprovechando al máximo los recursos disponibles en la plataforma.

```
Mon Aug 26 02:17:09 2024
+---------------------------------------------------------------------------------------+
| NVIDIA-SMI 535.104.05             Driver Version: 535.104.05   CUDA Version: 12.2     |
|-----------------------------------------+----------------------+----------------------+
| GPU  Name                 Persistence-M | Bus-Id        Disp.A | Volatile Uncorr. ECC |
| Fan  Temp   Perf          Pwr:Usage/Cap |         Memory-Usage | GPU-Util  Compute M. |
|                                         |                      |               MIG M. |
|=========================================+======================+======================|
|   0  Tesla T4                       Off | 00000000:00:04.0 Off |                    0 |
| N/A   61C    P8              11W /  70W |      0MiB / 15360MiB |      0%      Default |
|                                         |                      |                  N/A |
+-----------------------------------------+----------------------+----------------------+

+---------------------------------------------------------------------------------------+
| Processes:                                                                            |
|  GPU   GI   CI        PID   Type   Process name                            GPU Memory |
|        ID   ID                                                             Usage      |
|=======================================================================================|
|  No running processes found                                                           |
+---------------------------------------------------------------------------------------+
```

Figura 8. Descripción y verificación de tarjetas gráficas T4 y soporte para procesamiento CUDA en GPU

5.2. Obtención de documentos procesables

Durante este proceso, se confirmó que las tecnologías *open source* existentes ofrecían soluciones viables para la *diarización* de audios mediante el análisis de espectrogramas. En particular, se identificó que el modelo **NeMo MSDD (Multi-Scale Diarization Decoder)** utiliza los espectrogramas como representación visual de la energía de la señal de audio en diferentes frecuencias a lo largo del tiempo, permitiendo capturar tanto características de frecuencia como temporales críticas para la identificación de hablantes. Este modelo implementa un enfoque multiescala que combina información obtenida de ventanas de tiempo de diferentes longitudes (2.0s, 1.5s, 1.0s y 0.5s) para extraer características representativas y precisas de cada hablante. Como se ilustra en la Figura 9, el modelo evalúa diferentes escalas temporales para capturar tanto la calidad de las características del hablante como la granularidad temporal necesaria para decisiones precisas. Este proceso se logra mediante la estimación dinámica de pesos para cada escala, utilizando un mecanismo basado en redes neuronales convolucionales (CNN) o atención, dependiendo de la configuración, y borda de manera eficiente el compromiso entre la fidelidad de las representaciones del hablante y la resolución temporal [38].

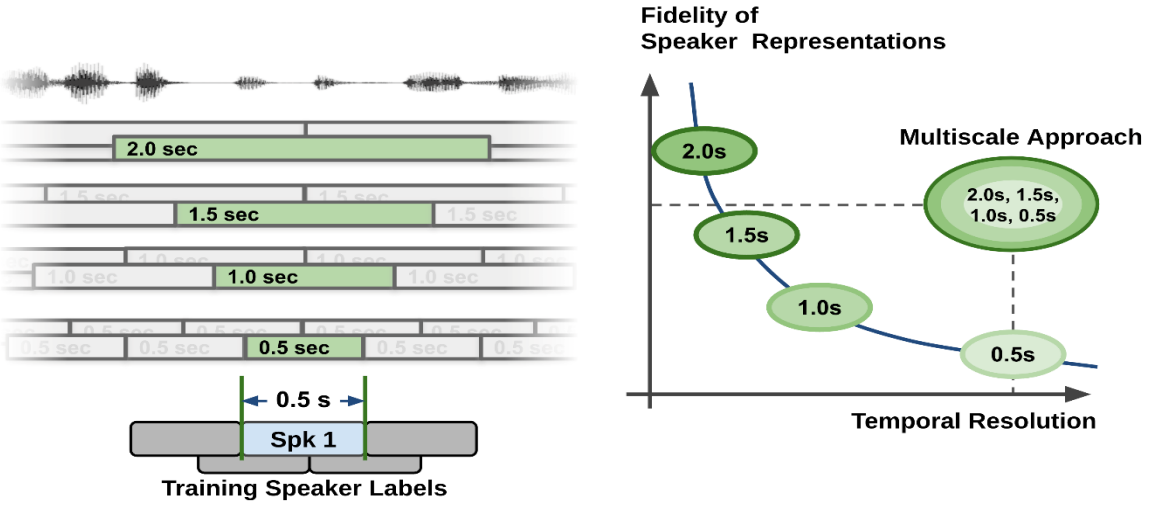


Figura 9. Representación multiescala de hablantes: resolución temporal y fidelidad

El flujo completo del modelo, representado en la Figura 10, incluye varias etapas clave que aseguran una *diarización* precisa y eficiente. Inicialmente, la entrada de audio se segmenta en múltiples escalas temporales, y los *embeddings* de hablantes se extraen mediante el modelo *TitaNet*. Estos *embeddings* se agrupan en clústeres iniciales que representan las diferentes voces presentes en el audio. Posteriormente, se calculan pesos dinámicos para cada escala utilizando similitud coseno, lo que permite generar vectores de contexto que reflejan con precisión la importancia de cada escala.

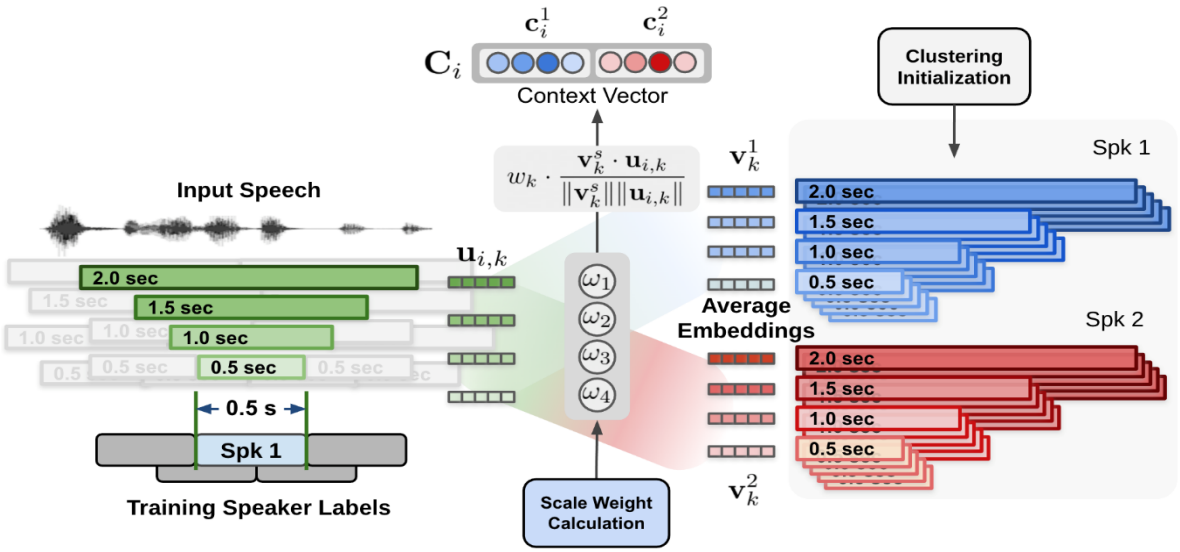


Figura 10. Proceso de representación multiescala y agrupamiento de hablantes

Finalmente, un modelo LSTM procesa estos vectores de contexto para producir etiquetas de hablantes con una resolución temporal fina de hasta 0,25 segundos por unidad de decisión, lo que garantiza una identificación precisa de los cambios de hablante, incluso en conversaciones con superposición de voces o segmentos breves. Este enfoque, ilustrado en la Figura 11, resulta especialmente valioso para generar transcripciones exactas, fundamentales para un análisis de sentimientos detallado y fiable. Además, un paso crítico para garantizar el correcto funcionamiento del modelo fue la transformación de los archivos de audio de estéreo a mono, ya que esta configuración, optimizada para señales de un solo canal, asegura que las características extraídas sean consistentes y adecuadas para el análisis multiescala.

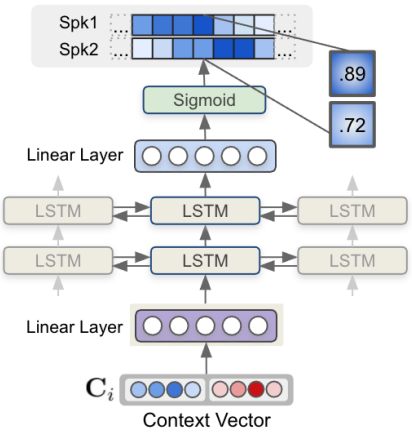


Figura 11. Arquitectura basada en LSTM para predicción de hablantes.

El resultado del proceso fue la obtención de 1.116 transcripciones, en las cuales se identificó y separó el texto correspondiente a cada hablante según sus turnos de intervención, quedando de esta forma un total de 41 audios que no fueron transcritos, al auditar estos archivos se determinó que en su mayoría no existe un ejercicio de conversación, por ejemplo, existen audio con 0 segundos de duración y otros en donde no hubo una respuesta por parte del *Speaker 1*, es decir, el cliente. Este enfoque permitió una organización clara y precisa de las interacciones [ver Tabla 2].

Tabla 2. Detalle de transcripciones realizadas

Folder	Cantidad WAV	Cantidad TXT	Audios no procesables
2023-04-01a	50	49	1
2023-04-10a	57	54	3
2023-04-11a	81	80	1
2023-04-15	72	69	3
2023-04-22a	74	74	0
2023-04-29a	56	50	6
2023-05-06a	89	78	11
2023-05-13a	77	68	9
2023-05-20a	80	80	0
2023-06-03a	77	76	1
2023-06-10a	59	58	1
2023-07-08a	108	106	2
2023-08-10a	122	119	3
2023-08-17a	51	51	0
2023-08-19a	48	48	0
2023-08-26a	56	56	0
Total general	**1.157**	**1.116**	**41**

Se puede observar en la Figura 12, que se generó un archivo TXT individual para cada uno de los audios procesados. Estos archivos fueron almacenados en una carpeta de Google Drive, utilizando una cuenta creada específicamente para este proyecto. Este método de almacenamiento no solo garantizó la seguridad y accesibilidad de los datos, sino que también facilitó el trabajo colaborativo entre los integrantes del equipo.

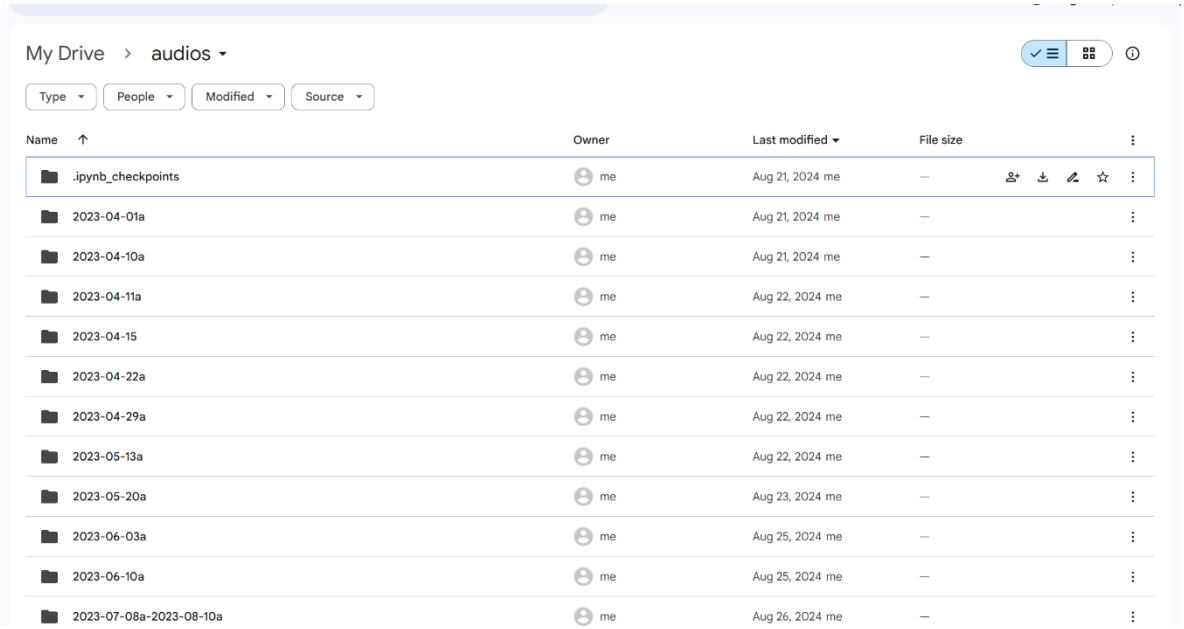


Figura 12. Repositorio archivos originales transcritos Google Drive

5.3. Limpieza de los datos

Una vez procesadas las transcripciones, se inició la etapa de limpieza y preprocesamiento de textos, un paso crucial en el flujo de trabajo para garantizar la calidad, consistencia y utilidad de los datos textuales. Este proceso fue diseñado e implementado utilizando un entorno colaborativo en Google Colab, lo que permitió un acceso eficiente a recursos computacionales y la integración con herramientas de almacenamiento en la nube, como Google Drive. Adicionalmente, se instalaron y configuraron bibliotecas especializadas de procesamiento de lenguaje natural (NLP), *incluyendo spaCy, unidecode, pyspellchecker*, y *pandas*, entre otras.

En primer lugar, se llevó a cabo un proceso de normalización del texto. Este paso incluyó la eliminación de caracteres especiales, tildes y símbolos no deseados mediante el uso de la biblioteca *unidecode* [39]. Además, se transformaron todos los textos a minúsculas para evitar inconsistencias derivadas de diferencias en el uso de mayúsculas y minúsculas. Este tipo de normalización es fundamental para garantizar que las palabras sean reconocidas como iguales, independientemente de su formato original [40].

Posteriormente, se aplicaron técnicas de corrección ortográfica utilizando la biblioteca *pyspellchecker*. Esto permitió identificar y corregir errores ortográficos comunes presentes en las transcripciones, incrementando significativamente la calidad del texto procesado. Este paso no solo mejora la comprensión general de los datos, sino que también

resulta esencial para etapas posteriores, como el análisis de sentimientos o la clasificación temática [41].

El procesamiento continuó con la *tokenización* y lematización de los textos, empleando el modelo de *spaCy* para español (*es_core_news_sm*), una herramienta ampliamente reconocida en proyectos de procesamiento de lenguaje natural [42]. La *tokenización* permitió dividir el texto en unidades más pequeñas denominadas tokens, como palabras o frases, mientras que la lematización transformó cada palabra en su forma base o raíz, como se recomienda en la literatura sobre PLN. Este enfoque redujo redundancias léxicas, unificando variaciones de una misma palabra y facilitando un análisis más coherente y eficiente [43].

Asimismo, se llevó a cabo la eliminación de palabras vacías, conocidas como *stop words*. Estas son palabras que, aunque son comunes en los textos, como artículos, preposiciones o pronombres, no aportan valor semántico significativo al análisis. Al eliminarlas, se logró enfocar el procesamiento en términos más relevantes y útiles para las siguientes fases del proyecto.

Además de estas transformaciones textuales, se realizó un análisis de frecuencia de palabras utilizando herramientas de visualización como gráficos e histogramas generados con *matplotlib* y *seaborn*. Este análisis permitió identificar patrones importantes en las conversaciones, como la presencia recurrente de ciertas palabras clave que podrían indicar temas dominantes o actitudes específicas como se observa en la Figura 13.

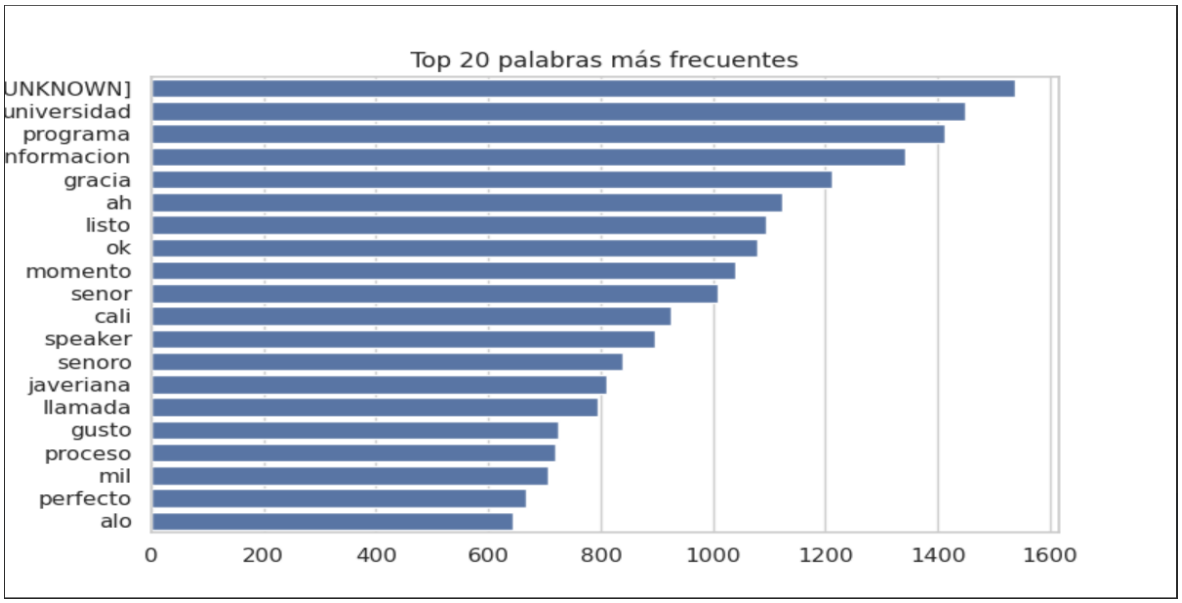


Figura 13. Top 20 palabras más frecuentes

5.4. Optimización de resultados mediante enfoques de preprocesamiento

Fue necesario explorar diferentes enfoques de limpieza dado que, como se aprecia en la Tabla 3, se expone la comparativa para la transcripción del registro identificado como *"bea64327334044e29f31d46fb3f2e42f_20230810t15_56_utc"* en donde se evidenció una notable pérdida de contexto que comprometió la calidad del corpus para el análisis de sentimientos. En el cuadro comparativo presentado, se observa cómo la aplicación de técnicas como la eliminación de *stopwords* y el *stemming* generaron una reducción significativa en la longitud del texto, pero también en su capacidad para expresar el tono y las emociones inherentes al discurso original. Por ejemplo, la frase "*mira lo que pasa es que ayer realicé un pago y no me han dado la notificación*" fue transformada en "*mirar pasar ayer realizar pago notificación*", eliminando palabras clave como "*no*" y "*han dado*" que eran esenciales para comprender la insatisfacción del cliente. Este fenómeno de simplificación excesiva evidenció la necesidad de ajustar las estrategias de limpieza para evitar comprometer el contexto semántico de los datos.

Tabla 3. Comparativo de transcripciones

Transcripción original	**Transcripción preprocesada**
Speaker 0: buenos días buenos días.	Speaker 0:
Speaker 1: mira lo que pasa es que ayer realicé un pago y no me han dado la notificación. no sé si ya llegó el correo o no. necesitaba saber qué pasaba.	Speaker 1: mirar pasar ayer realizar pago notificacion llego correo necesitar pasar
Speaker 0: me puedes indicar tu número de documento por favor?	Speaker 0: poder indicar número documento favor
Speaker 1: uno ciento doce cuatro ochenta y cuatro cinco noventa y cuatro Ginette Carolina Mina. Correcto.	Speaker 1: ciento ochenta noventa Ginette carolina mina correcto
Speaker 0: Ya estás, ya apareces inscrito pago en el programa. El correo ya, o sea, ya el correo donde te dan indicaciones para el inicio del programa te llega un día antes del inicio de la formación.	Speaker 0: apareces inscrito pago programa correo correo indicación inicio programa llegar inicio formación
Speaker 1: Ah, ok, ok. Alrededor. Sí, sí había subido el pago, ¿no?	Speaker 1: ah ok ok subido pago
Speaker 0: Sí, es correcto, el pago fue inscrito, inclusive, eh, Veo que está montado el recibo ya cancelado con el timbre de la máquina en Banco de	Speaker 0: correcto pago inscribir inclusive eh veo montado recibo cancelado timbre

Transcripción original	**Transcripción preprocesada**
Occidente y realizaste el pago, según veo. Sí, sí, correcto.	maquina banco occidente realizaste pago veo correcto
Speaker 1: Yo envié los soportes.	Speaker 1: enviar soporte
Speaker 0: Es correcto. Nosotros, pues, por lo mismo solicitamos el soporte para poderlo dejar en el mismo sistema dentro del usuario. por si de pronto ocurre alguna no sincronización del pago, nosotros poderla solicitar de manera interna. Pero sí, ya aparece registrado tu pago, ya aparece inscrita paga en el programa, lo que indica que estaríamos contando con tu participación dentro del mismo. Y lo que te indico, el día anterior al inicio de la formación te llegará un correo electrónico por parte de logística de la universidad, donde te dan todas las indicaciones para el inicio del programa. Este se envía aproximadamente a las cuatro de la tarde, alrededor de las cuatro de la tarde. Si no te llega... Ese día antes nos escribes el mismo día de la formación en horas de la mañana para poderte compartir el correo con la información, ya que logística nos copia a nosotros.	Speaker 0: correcto solicitar soporte dejar sistema usuario ocurrir sincronización pago solicitar interno aparecer registrado pago aparecer inscrito paga programa indicar estareír contar participación indicar inicio formación llegar correo electrónico logística universidad indicación inicio programa enviar llegar escribes formación horas mañana poderte compartir correo información logística copiar
Speaker 1: Ok, muchas gracias.	Speaker 1: ok gracia
Speaker 0: Vale, con todo gusto, señorita Ginette. Que estés muy bien, hasta luego.	Speaker 0: vale gusto señorita Ginette estes
Speaker 1: Hasta luego.	Speaker 1:

Por tanto, para abordar la problemática de la pérdida de contexto y garantizar la calidad del corpus, se implementaron ajustes en el flujo de preprocesamiento que contemplaron múltiples enfoques, permitiendo analizar sus efectos sobre la retención semántica del texto. Se desarrollaron variantes que incluyeron: normalización básica, lematización, *stemming*, eliminación de *stopwords* y combinaciones de estas técnicas. Este enfoque no solo buscó reducir el ruido en las transcripciones, sino también preservar los elementos lingüísticos esenciales para identificar emociones y clasificar sentimientos. En la Figura 14 se observa una muestra aleatoria que sigue un patrón en donde los enfoques que más presentan reducción de longitud sobre los textos transcritos son justamente aquellos que presentan combinaciones entre sí.

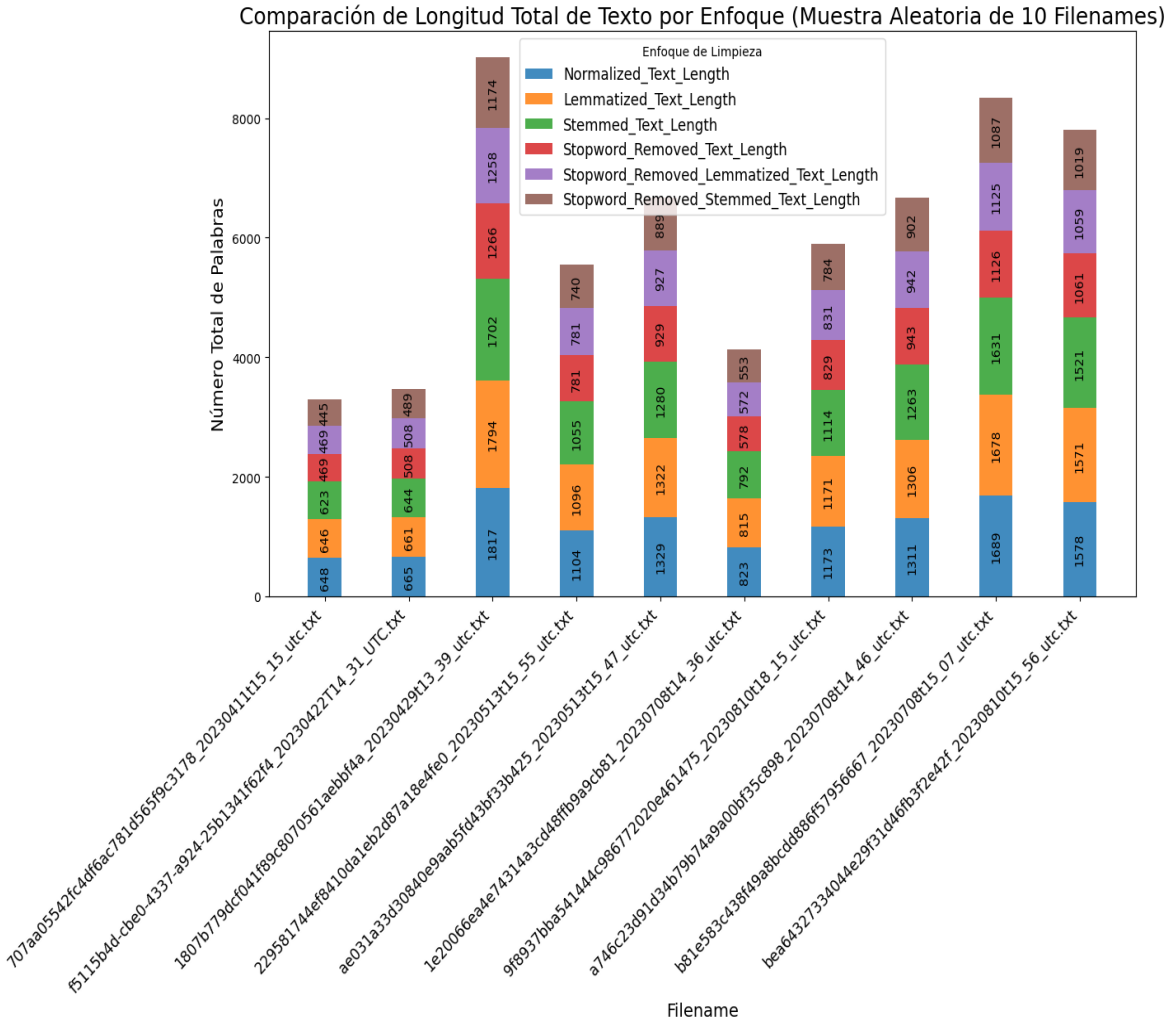


Figura 14. Comparación de Longitud (Muestra aleatoria)

La comparación entre los distintos métodos reveló que la combinación de lematización con retención de *stopwords* ofrecía el mejor equilibrio entre limpieza y contexto, proporcionando un corpus que permite explorar y analizar en los siguientes capítulos, diversos métodos de clasificación. En la Figura 15 se observa un mapa de calor que refleja cómo los distintos métodos de preprocesamiento impactan en la cantidad de texto retenido en una muestra aleatorios. Esto permite identificar qué métodos pueden ser más efectivos para reducir el ruido textual, pero también evidencia posibles pérdidas de información si la reducción es excesiva, especialmente en textos más cortos.

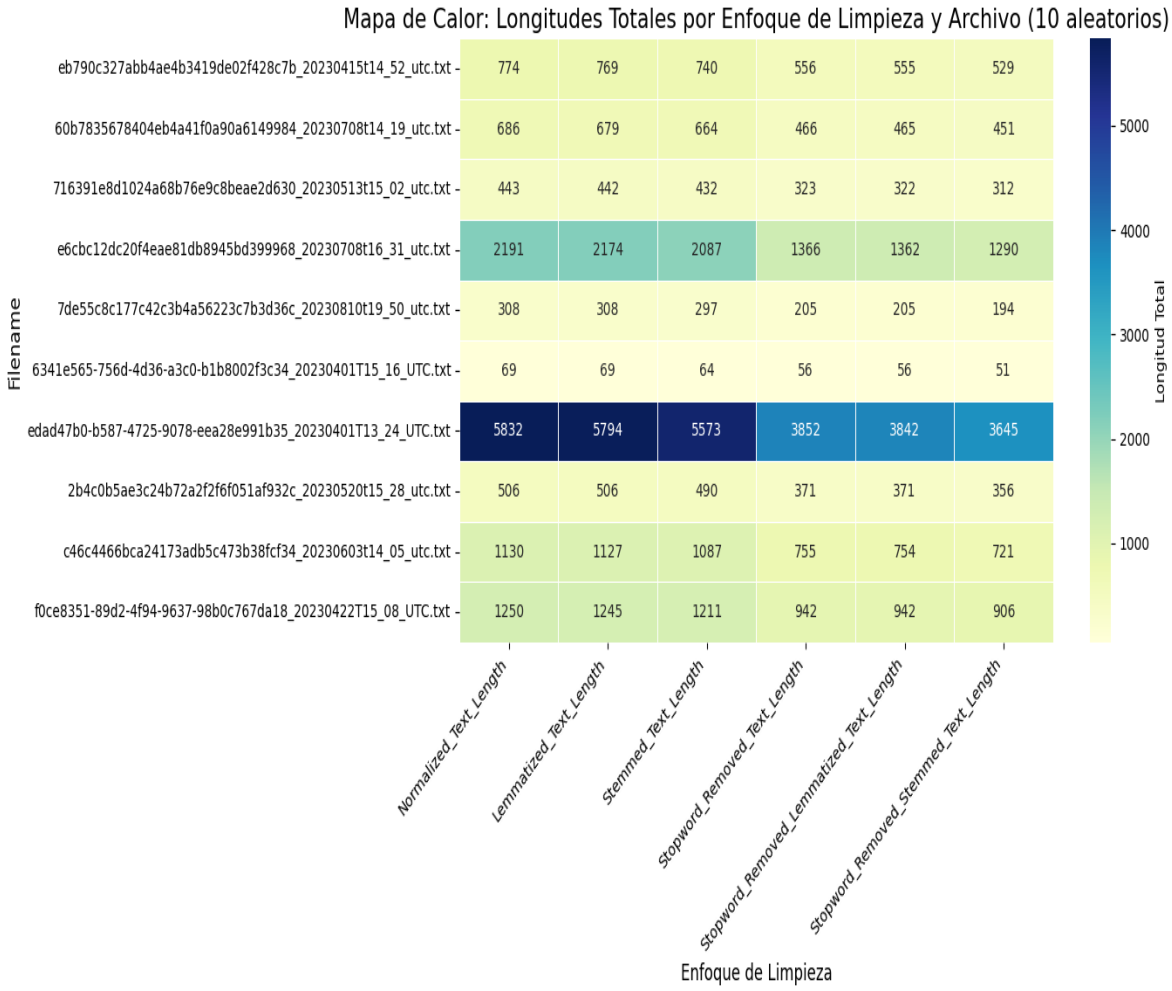


Figura 15. Mapa de calor por longitudes y enfoque de limpieza

En la Figura 16 se observa el promedio de retención por método de preprocesamiento, calculado como la relación entre la longitud resultante de cada enfoque y la longitud de la transcripción original de forma global. Los resultados muestran que los enfoques que conservan una mayor longitud del texto son, en orden descendente: **Normalización**, **Lematización** y **Stemming**, todos ellos sin la eliminación de *stopwords*. Esto indica que los métodos menos agresivos en la reducción de texto retienen más información clave para el análisis esperado.

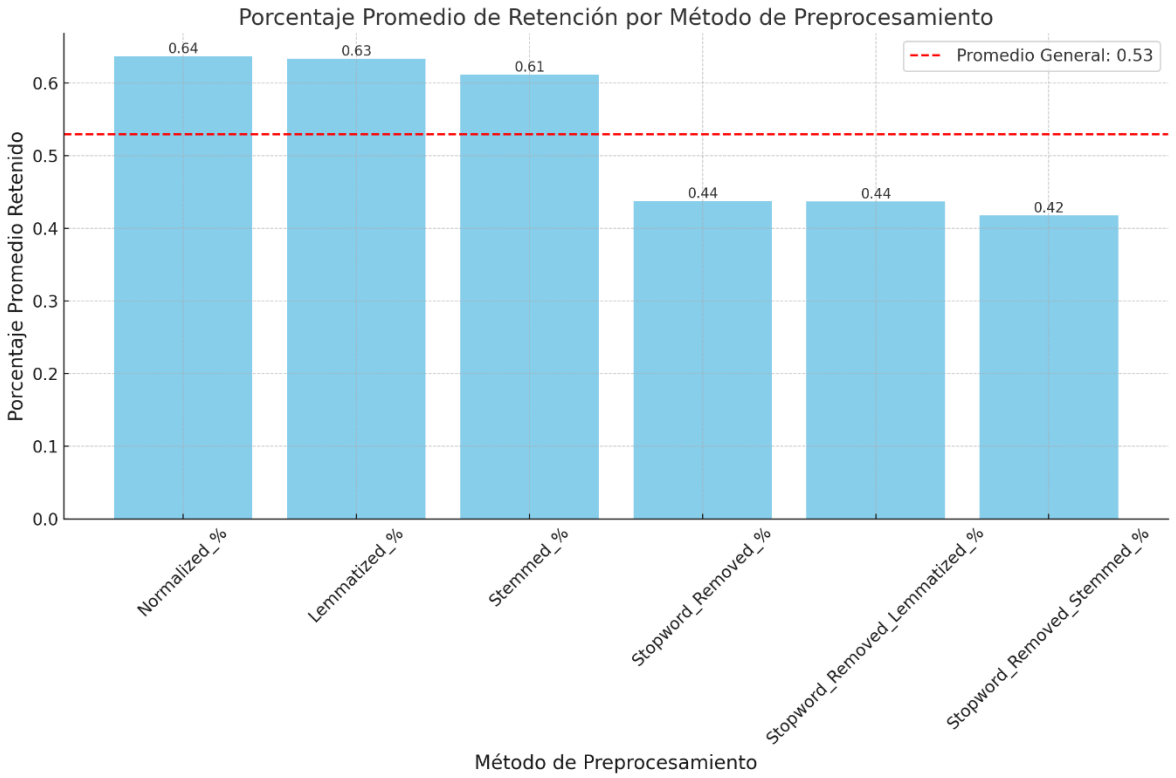


Figura 16. Promedio de retención por método de preprocesamiento

Por último, para aprovechar los datos analizados, se construyó un dataset con las siguientes dimensiones: Folder, Filename, Speaker, Text, Text_Length, Normalized_Text, Lemmatized_Text, Stemmed_Text, Stopword_Removed_Text, Stopword_Removed_Lemmatized_Text, Stopword_Removed_Stemmed_Text, Normalized_Text_Length, Lemmatized_Text_Length, Stemmed_Text_Length, Stopword_Removed_Text_Length, Stopword_Removed_Lemmatized_Text_Length y Stopword_Removed_Stemmed_Text_Length. Adicionalmente, se desarrolló una función que genera un archivo TXT para cada preprocesamiento, organizado con la misma estructura de carpetas y nombres de archivo originales. Este enfoque permite aprovechar los datos procesados en etapas posteriores, facilitando su implementación en un producto mínimo viable orientado al uso por parte del cliente final.

6. ORGANIZACIÓN Y ETIQUETADO DEL CORPUS DE TRANSCRIPCIONES PARA EL ENTRENAMIENTO DEL MODELO DE ANÁLISIS DE SENTIMIENTOS

En este apartado se detalla la tercera fase de CRISP-DM, denominada *Preparación de los datos*, cuyo objetivo principal es transformar y estructurar los datos disponibles en un formato adecuado para su análisis posterior. Esta etapa es esencial para garantizar que los datos sean consistentes, relevantes y estén optimizados para el proceso de modelado [37].

6.1. Definición de criterios de etiquetas

Tras la creación del *dataset* a partir de los distintos enfoques de preprocesamiento descritos en el capítulo anterior, se definieron en primer lugar los criterios de etiquetado —positivo, neutral o negativo— atendiendo al contexto de gestión comercial-servicio de la Pontificia Universidad Javeriana Cali, en el que un agente contacta a personas interesadas en sus ofertas académicas. En las interacciones clasificadas como **positivas**, el interlocutor demuestra entusiasmo, formula preguntas detalladas, utiliza expresiones como *"¡Qué interesante!*" o "*Me encantaría saber más*", acepta programar citas y agradece el contacto con un tono amable y entusiasta. Las **neutrales** se caracterizan por respuestas breves y profesionales ("*Entiendo*", "*De acuerdo*"), ausencia de emociones intensas, indecisión o solicitud de tiempo para pensar, y una interacción cordial pero limitada. Finalmente, las **negativas** incluyen manifestaciones de desinterés o rechazo explícito ("*No me interesa*", "*No, gracias*"), críticas a la oferta o la institución, un lenguaje cortante y un tono de voz serio o impaciente que busca concluir la llamada rápidamente [ver Tabla 4].

Tabla 4. Categorías para el etiquetado de transcripciones

Categoría	**Criterios**
Positivo	Muestra entusiasmo por la oferta académica
	Hace preguntas detalladas o solicita más información
	Utiliza frases como '¡Qué interesante!', 'Me encantaría saber más', etc.
	Acepta programar cita, se muestra dispuesto a inscribirse, compartir la información
	Tono de voz amable, entusiasta; ríe o muestra alegría.
	Agradece la llamada o contacto
Neutral	Escucha sin mostrar emociones fuertes
	Responde con afirmaciones simples ('Entiendo', 'De acuerdo')
	Indecisión, necesita tiempo para pensar
	Tono de voz estable y profesional
	No utiliza lenguaje emocional ni muestra entusiasmo/descontento
	Interacción cordial pero limitada

Categoría	Criterios
Negativo	Expresa desinterés o rechazo
	Frases como 'No me interesa', 'No, gracias'
	Critica la oferta o institución
	Muestra frustración, molestia, lenguaje cortante
	Tono de voz serio, molesto, impaciente; intenta finalizar rápido

6.2. Etiquetado del corpus

El etiquetado del corpus constituye una etapa crítica para el análisis de sentimientos, ya que define la semántica y la estructura sobre la que se entrenan los modelos [44]. En este proyecto, el etiquetado fue diseñado para capturar las emociones predominantes en cada intervención de la interacción telefónica, facilitando así la evaluación de la experiencia del usuario y la clasificación de sentimientos en categorías relevantes. Además, establecer un estándar de calidad mediante un etiquetado riguroso es esencial para garantizar la confiabilidad de los análisis subsiguientes [45].

De esta forma, el etiquetado manual fue llevado a cabo por los tres integrantes del proyecto de forma independiente, donde analizamos la totalidad de la muestra de las transcripciones. Este proceso permitió establecer una línea base de calidad y fue diseñado de manera rigurosa para garantizar la precisión de los resultados asignando una etiqueta (positiva, negativa o neutra) a cada una de ellas. Este enfoque tuvo como objetivo evitar sesgos grupales y asegurar que cada transcripción recibiera tres interpretaciones independientes.

Es importante destacar que la etiqueta *Neutral* fue la predominante en el conjunto de datos. Esto se debe a que, en la lógica del negocio, durante la mayor parte de la conversación, las intervenciones de los interlocutores se centran en intercambios informativos y protocolos propios del contexto, sin manifestar patrones emocionales claros. Sólo en los momentos de cierre de la llamada —cuando el interlocutor define su disposición— se tiende a asignar las etiquetas *Positiva* o *Negativa*. Al evaluar la distribución de clases en el *dataset*, se identificó un desequilibrio que deberá abordarse en etapas futuras del entrenamiento mediante técnicas de remuestreo para lograr un balance de clases, entre otras, para mejorar la capacidad del modelo de clasificar de una forma precisa.

Para efecto de visualización de este proceso se tomó aleatoriamente la transcripción con el id: *0c1dacc2-0d21-474f-9f30-0b2bc3069d1a_20230401T16_11_UTC.txt* de la cual se presenta un extracto de la conversación [ver Tabla 5].

Tabla 5. Ejemplo etiquetado de transcripción

Folder	Speaker	Andrea	Daniel	Jhon	Text
2023-04-01a	Speaker 0	Neutral	Neutral	Neutral	Buen día, le habla Vanessa.
2023-04-01a	Speaker 1	Positivo	Positivo	Positivo	¿En qué le puedo ayudar? Hola, buenos días. Habla Joan Camilo. Perdona la bulla. Lo que pasa es que a mí me interesaría ingresar a estudiar enfermería.
2023-04-01a	Speaker 0	Neutral	Neutral	Neutral	Dime, ya terminaste. Dime.
2023-04-01a	Speaker 1	Positivo	Neutral	Positivo	Perdón. Este, me gustaría empezar a estudiar enfermería, yo ya tengo bachiller, yo tengo dos títulos, bueno, un título, estoy cursando el último, estoy haciendo las prácticas, pero pues yo quisiera como averiguar con tiempo, ¿me hago entender?
2023-04-01a	Speaker 0	Neutral	Neutral	Neutral	Sí, ¿y qué deseas saber?
2023-04-01a	Speaker 1	Positivo	Positivo	Neutral	Yo quiero estudiar enfermería, pero pues mis recursos, la verdad, son como un poco limitados. y a mí me llegaron a hablar sobre una beca, algo que se llama como la generación E. Tengo como dos, tres compañeros allá en donde estoy haciendo mis prácticas que me hablaron de eso, que ellos estudiaron en diferentes universidades. y pues yo estuve mirando como en las páginas y sale la opción de esa beca, como inscribirse a eso. Yo la verdad no conozco muy bien sobre el proceso, ellos pues no me explicaron bien.
2023-04-01a	Speaker 0	Neutral	Neutral	Neutral	¿A qué número marcaste? ¿Cuál fue el número? Sí, en este momento, ¿qué número marcaste?
2023-04-01a	Speaker 1	Neutral	Neutral	Neutral	Un seis cero dos tres noventa y ocho once cuarenta y siete, porque ya me hago como un tres veintiuno, creo. También que me salió en la plataforma, pero no contestan nada, o sea, no, nada.
2023-04-01a	Speaker 0	Neutral	Neutral	Neutral	Debes de comunicarte por ese medio para que te puedan transferir al área financiera, que en eso, que te pueden brindar información sobre esta beca.
2023-04-01a	Speaker 1	Positivo	Positivo	Neutral	Una pregunta, ¿qué me podrían dar información sobre la carrera en sí? Sí. La parte de la beca. ¿Me puedes, por favor, dar información para saber más o menos también?

6.3. Definición de etiqueta final

Una vez completadas las etiquetas individuales, se aplicó el método de voto por mayoría para determinar la etiqueta final de cada interacción entre el agente y el usuario. En este esquema, cada revisor "vota" marcando una única categoría (Positivo, Neutral o Negativo) y gana la etiqueta que obtenga al menos dos de los tres votos. Este procedimiento garantiza una decisión democrática y refleja con claridad la preferencia mayoritaria del equipo, este proceso se llevó a cabo de forma ágil en una hoja de cálculo para hallar la columna del sentimiento final mediante el siguiente paso a paso con una función en Excel:

CONTAR.SI(C2:E2; "Positivo") >= 2

*– Cuenta cuántas de las tres celdas (Andrea, Daniel, Jhon) contienen la palabra "**Positivo**".*

– Si el resultado es 2 o 3, significa que al menos dos revisores votaron "Positivo".

SI(...; "Positivo"; ...)

– Si la condición anterior se cumple, la función devuelve "Positivo" directamente.

CONTAR.SI(C2:E2; "Neutral") >= 2

*– Si no hubo mayoría "Positivo", esta segunda comprobación cuenta las celdas con "**Neutral**".*

– Si al menos dos revisores votaron "Neutral", la condición es verdadera.

SI(...; "Neutral"; "Negativo")

– Si la mayoría es "Neutral", devuelve "Neutral".

*– Si tampoco hay mayoría "Neutral" (lo que implica que hubo al menos dos votos "**Negativo**"), devuelve "Negativo".*

Visto de una forma gráfica, en la imagen [ver Figura 17] se representa como un árbol de decisión la secuencia de operaciones que se realizaran para la obtención de la etiqueta final.

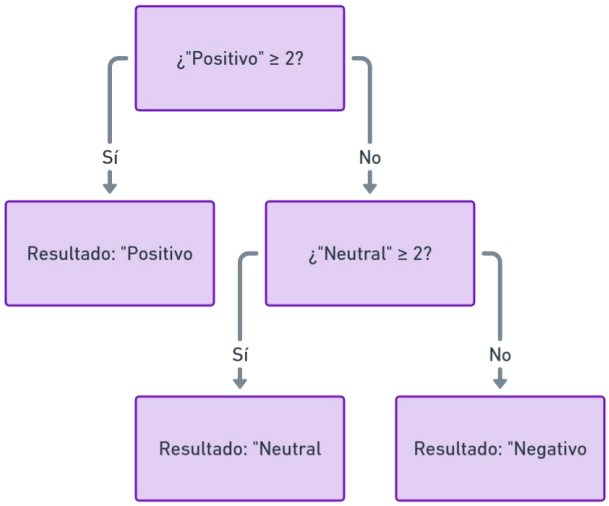


Figura 17. Árbol de decisión para obtener etiqueta final

El análisis de consistencia entre los evaluadores se llevó a cabo mediante el cálculo del coeficiente de concordancia *Kappa de Fleiss*, una medida estadística utilizada para evaluar la consistencia de clasificación en tareas con múltiples evaluadores [4].

$$\kappa = \frac{P_o - P_e}{1 - P_e}$$

Donde P_o es la Proporción de concordancia observada entre los evaluadores y P_e es la proporción de concordancia esperada por azar, ver Figura 18.

```
{'Po (Concordancia observada)': np.float64(0.9801400863923453),
 'Pe (Concordancia esperada por azar)': np.float64(0.7688911110272294),
 'Kappa calculado manualmente': np.float64(0.9140668552562917)}
```

Figura 18. Concordancia observada vs esperada

En este estudio, los tres integrantes del proyecto etiquetaron manualmente intervenciones orales transcritas en categorías emocionales (Positivo, Neutral, Negativo). El valor obtenido fue ***κ = 0,914*** [ver Figura 19], lo cual refleja un **alto nivel de acuerdo**, especialmente significativo si se considera la naturaleza del lenguaje conversacional y la lógica del negocio que subyace en este tipo de interacciones —principalmente centradas en la comunicación comercial, informativa y de servicio al cliente.

Este nivel de consistencia no solo valida la calidad del etiquetado, sino que establece un marco sólido para el entrenamiento y validación de modelos de análisis de sentimientos.

El etiquetado manual, además de proporcionar un conjunto confiable de datos, permitió detectar patrones de expresión emocional y ambigüedades contextuales que pueden ser abordadas en etapas posteriores del proyecto.

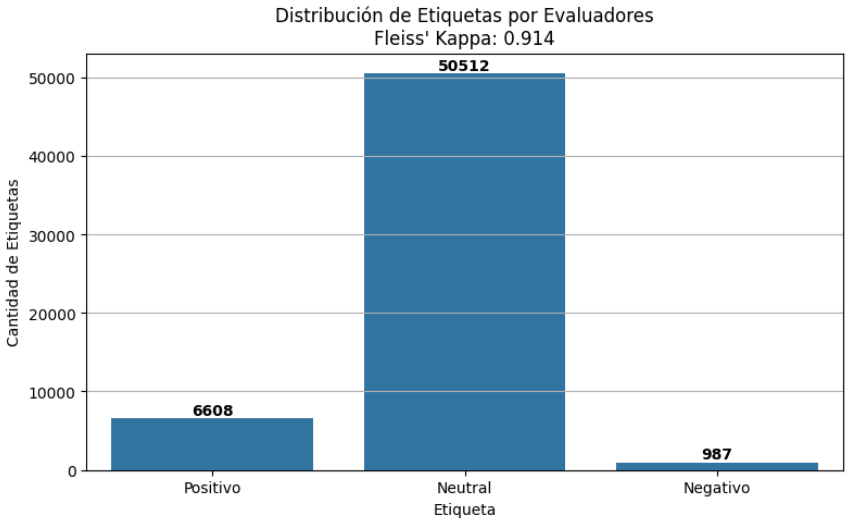


Figura 19. Distribución de etiquetas por evaluadores

El etiquetado manual no solo proporcionó un conjunto de datos confiables para validar el desempeño del modelo, sino que también permitió identificar patrones y ambigüedades que podrían ser abordadas en las etapas posteriores, ver Figura 20.

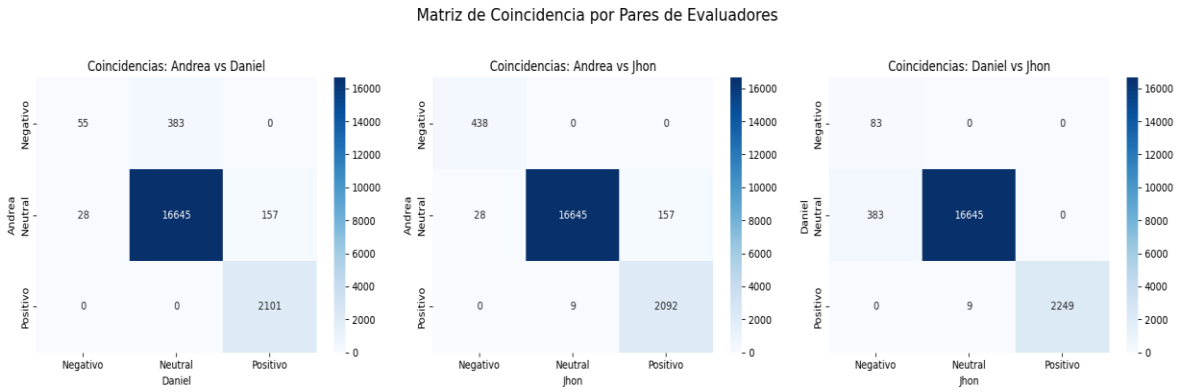


Figura 20. Matriz de coincidencia por pares

Este proceso ayudó a establecer un estándar de calidad, asegurando que el análisis de sentimientos reflejara con precisión las emociones expresadas en las transcripciones.

7. IMPLEMENTACIÓN DE MODELOS DE PROCESAMIENTO DE LENGUAJE NATURAL PARA EL ANÁLISIS DE SENTIMIENTOS EN TRANSCRIPCIONES DE LLAMADAS

En este apartado se aborda la cuarta fase del modelo CRISP-DM, denominada *Modelado*, la cual contempla la selección, configuración y entrenamiento de algoritmos con el propósito de alcanzar los objetivos establecidos en el proyecto. Esta etapa implica un proceso iterativo de pruebas y ajustes para optimizar el desempeño de los modelos. En este contexto, se da cumplimiento al tercer objetivo específico de este proyecto aplicado.

El análisis de sentimientos aplicado a conversaciones de atención al cliente representa un desafío que requiere no solo la interpretación del lenguaje natural de manera precisa, sino también la comprensión de su contexto emocional y semántico. Este tipo de análisis demanda un enfoque que combine tecnología avanzada con metodologías estructuradas para garantizar resultados confiables y accionables. Por tanto, se exploraron diferentes perspectivas metodológicas con el fin evaluar cómo cada modelo maneja las complejidades del análisis de sentimientos.

7.1. Selección y descripción de los modelos implementados

La selección de los modelos adecuados es una etapa crítica en cualquier proyecto de análisis de datos, debido a la influencia directa que esta decisión tiene sobre la calidad de los resultados, la eficiencia del proceso y la interpretabilidad de los análisis. Este proceso implica la elección de algoritmos y arquitectura y la evaluación de sus capacidades para abordar las necesidades específicas del problema, en este caso, la clasificación de emociones en interacciones telefónicas del corpus.

El objetivo principal de esta fase fue explorar enfoques que, combinados, proporcionaran un análisis robusto, preciso, contextualizado y eficiente del sentimiento expresado por los usuarios. Para lograrlo, se adoptó una estrategia comparativa que incluyó modelos basados en arquitecturas de redes neuronales profundas, modelos híbridos y métodos tradicionales enriquecidos con representación vectorial de texto, a continuación, se representan las arquitecturas de modelos empleados [ver Tabla 7].

Tabla 6. Descripción de arquitecturas a evaluar

Enfoque	Representación de entrada	Arquitectura principal	Justificación metodológica
MLP con TF-IDF	TF-IDF (uni + bi-gramas, 5 000 dimensiones)	Perceptrón multicapa feed-forward	Línea base clásica; muy rápido e interpretable; mide peso de términos aislados.
TextCNN	Embeddings (300 dim) de vocabulario específico	CNN 1D (múltiples tamaños de filtro) + max-pool	Captura automáticamente n-gramas locales sin importar posición; eficiente en textos cortos.
Bi-LSTM	Embeddings (300 dim) + secuencia de longitud fija	Red LSTM bidireccional	Modela dependencias a largo plazo en ambas direcciones; esencial para matices contextuales.
Bi-GRU	Igual que Bi-LSTM	Red GRU bidireccional	Variante más ligera de LSTM; similar capacidad de memoria con menos parámetros.
CNN - LSTM híbrido	Embeddings (300 dim) + convolución 1D local	CNN 1D seguida de LSTM	Une detección local de patrones (CNN) con lectura secuencial (LSTM).
DistilBETO fine-tuned	Tokenización WordPiece, secuencia hasta 100 tokens	DistilBERT (6 capas) + capa de clasificación	Representación contextual profunda; aprovecha pre-entrenamiento en español.

Cada uno de estos enfoques aporta una perspectiva distinta: desde modelos muy rápidos y simples (MLP-TF-IDF) hasta arquitecturas profundas contextualizadas (DistilBETO). La descripción de sus hiper parámetros y la forma en que se combinaron se detalla a continuación.

7.2. Alistamiento del corpus para los experimentos

En este capítulo se describe la metodología empleada con el fin de evaluar de forma conjunta cómo influye los enfoques de preprocesamiento y el balance de clases. Como paso inicial se codificaron las etiquetas de sentimiento en valores numéricos para facilitar el entrenamiento supervisado. Cada categoría recibió un identificador entero según la siguiente [ver Tabla 7]

Tabla 7. Identificador numérico por clase

Sentimiento	Identificador
Negativo	0
Neutral	1
Positivo	2

Se empleó *LabelEncoder* de *scikit-learn* para transformar las etiquetas de sentimiento en valores enteros (variable *y_enc*), de modo que los modelos procesaran categorías discretas en lugar de cadenas de texto. De esta forma, se plantearon con dos objetivos preparatorios:

Primero, identificar la estrategia de balanceo de clases más efectiva, dado el predominio de la etiqueta "*Neutral*" (como se discutió en el capítulo 6.2) y el consecuente sesgo en la distribución de sentimientos. Segundo, seleccionar las representaciones textuales más discriminativas —TF-IDF, lematización y combinaciones de *n-gramas*— para optimizar el rendimiento de los modelos de aprendizaje profundo.

Luego, el proceso experimental continuó con la partición estratificada del conjunto de datos en un esquema 80/20 (entrenamiento/prueba). Sobre la partición de entrenamiento (*X_train_raw, y_train*), se aplicó una configuración común de vectorización mediante TF–IDF con las siguientes características:

- Tamaño del vocabulario: 5.000 características
- N-gramas incluidos: unigrama y bigrama (1 a 2)
- Tokenización estándar con normalización previa del texto

Posteriormente, sobre este espacio vectorizado, se aplicaron cuatro técnicas de muestreo definidas en *sampling_strategies*, con el fin de balancear la distribución de clases y mitigar el sesgo hacia la clase neutral:

- SMOTE *(Synthetic Minority Over-sampling Technique)*
- SMOTEENN (combinación de sobre y submuestreo)
- SMOTETomek
- RandomUnderSampler

Estas estrategias permitieron generar subconjuntos de entrenamiento artificialmente balanceados, manteniendo la codificación y dimensionalidad uniforme entre ellos.

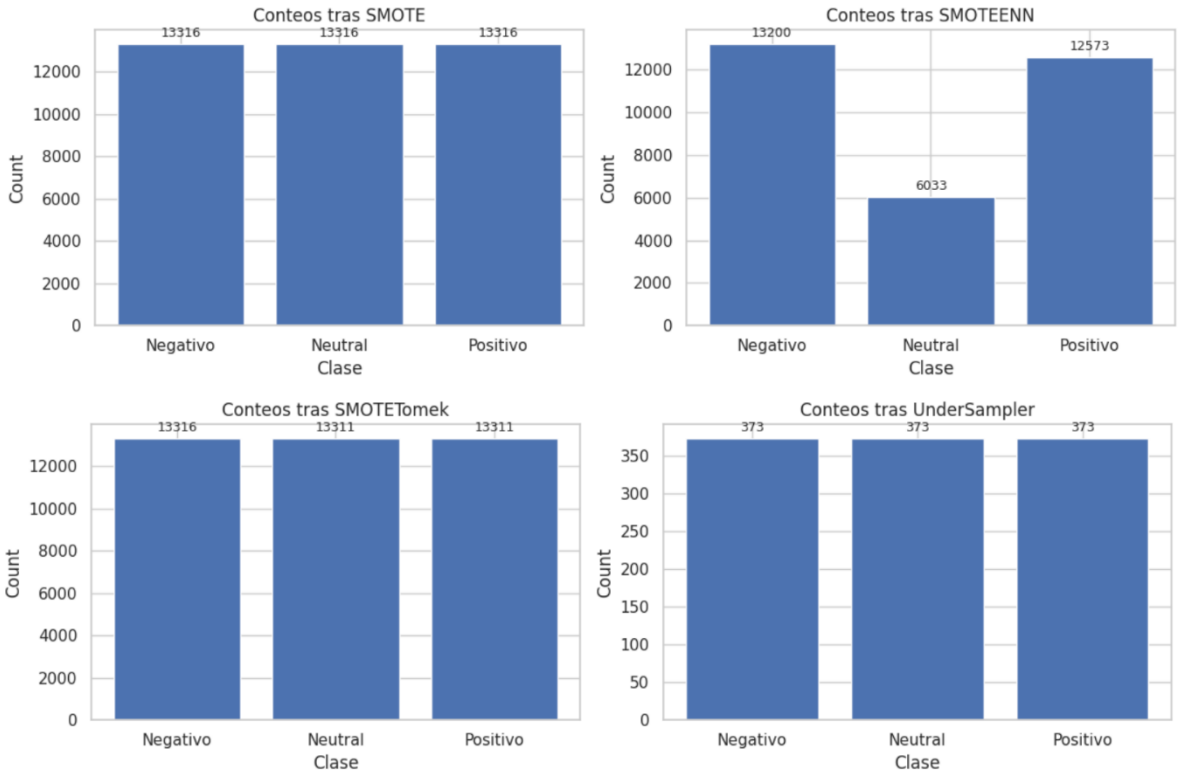


Figura 21. Distribución de cada técnica de balanceo

Tras aplicar *SMOTE* se obtuvo un balance perfecto (33 % para cada clase) al generar 13.316 muestras sintéticas en todas las clases, en *SMOTETomek* 13.316 en Negativo y 13.311 en *Neutral* y *Positivo*, en cambio *SMOTEENN* combina sobremuestreo y limpieza para producir un ligero desequilibrio (≈ 42 % negativo, 40 % positivo, 18 % neutral) y *UnderSampler* reduce todas las clases a 373 instancias sin crear nada nuevo. [ver Figura 21. Distribución de cada técnica de balanceo [Figura 21].

Una vez lograda una distribución equilibrada de clases mediante diversas estrategias de muestreo, el siguiente paso consistió en identificar cuáles versiones del texto preprocesado ofrecían mayor capacidad discriminativa para la tarea de clasificación de sentimientos. Dado que la efectividad de los modelos de aprendizaje profundo depende en gran medida de la calidad y expresividad de las representaciones de entrada, se consideró necesario evaluar comparativamente las diferentes variantes textuales generadas durante el preprocesamiento (original, normalizado, lematizado, con o sin *stopwords*, entre otras). Esta fase exploratoria permitió determinar, de forma rápida y sistemática, cuáles columnas presentaban una señal más clara de polaridad emocional cuando eran transformadas en vectores TF-IDF, y sirvió como filtro previo para seleccionar las representaciones más informativas que alimentarían los modelos más complejos en fases posteriores.

El proceso aplicado consistió en evaluar la capacidad predictiva de forma controlada y comparable de todas las columnas derivadas del texto original de la transcripción. En primer lugar, utilizando la partición estratificada del conjunto de datos, los textos fueron transformados mediante una vectorización TF–IDF configurada para capturar unigramas y bigramas, con un límite máximo de 5.000 características, ajustando el vectorizador únicamente sobre el conjunto de entrenamiento para evitar fugas de información. Con esta representación numérica, se entrenó un clasificador *Multinomial Naive Bayes* que, por su eficiencia computacional y robustez estadística, es especialmente adecuado para tareas de prototipado rápido sobre datos textuales dispersos. Una vez entrenado, el modelo fue evaluado sobre el conjunto de prueba, calculando métricas clásicas como la exactitud (*accuracy*), el macro-F1 score, el *recall* por clase y el área bajo la curva ROC macro (ROC-AUC). Además, se obtuvieron las probabilidades de predicción por clase para construir curvas ROC multiclase las cuales son presentadas a continuación, [ver Figura 22].

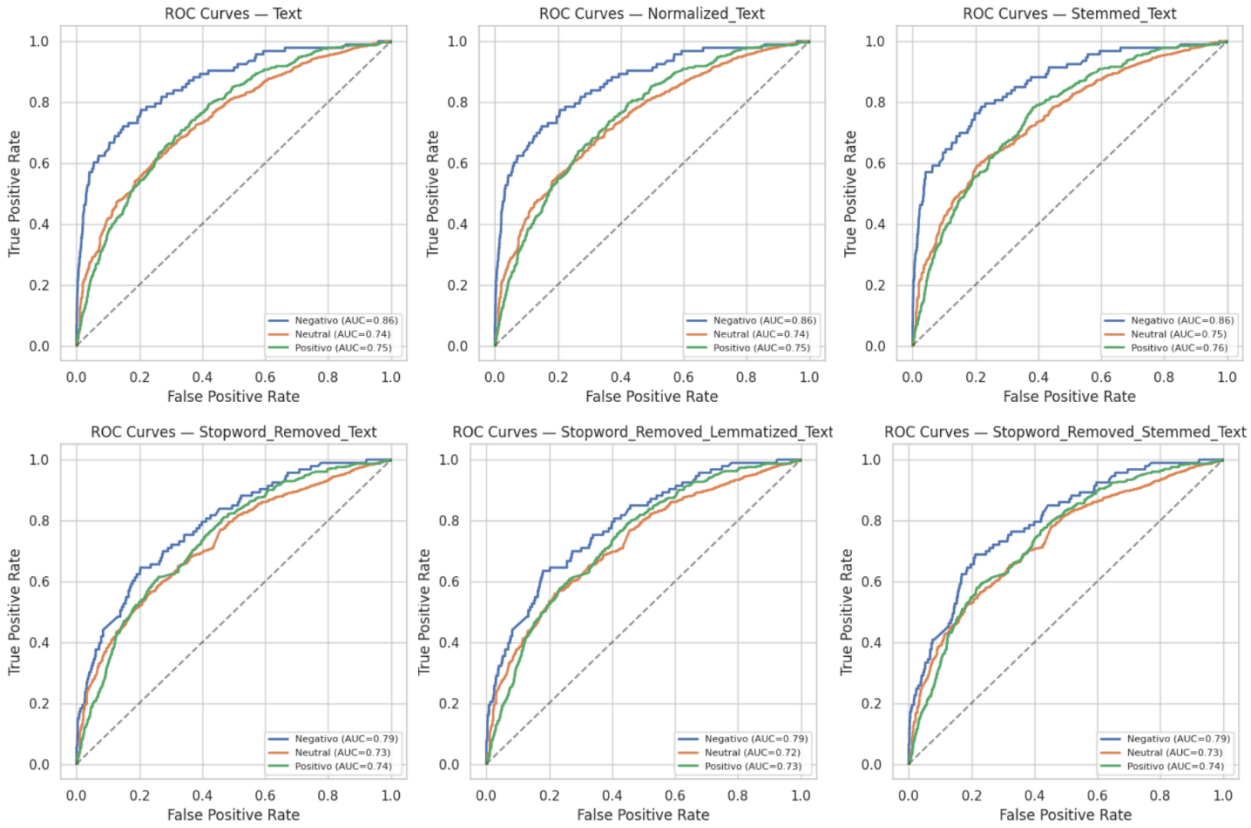


Figura 22. Curva ROC para cada variación de texto

De acuerdo con estos resultados se determinó que las columnas *Text*, *Normalized_Text*, *Lemmatized_Text* y *Stemmed_Text*, conservaron un rendimiento destacado en términos de precisión y capacidad discriminativa: la precisión promedio osciló en torno al 85,4–85,5 %, y el área bajo la curva ROC macro se aproximó a 0,79. En contraste, las

variantes con eliminación de *stopwords* presentaron una degradación significativa, con caídas tanto en precisión como en macro-F1 (~0,32 frente a ~0,35) y una reducción del ROC-AUC (~0,76 vs. ~0,79), lo que sugiere que en este dominio las palabras funcionales aportan información útil para la tarea de clasificación emocional. Estos hallazgos validan la hipótesis de que ciertas transformaciones lingüísticas —como la lematización y el *stemming*— potencian la capacidad del modelo para discriminar sentimientos, mientras que eliminar *stopwords* puede suprimir información contextual relevante, particularmente en interacciones orales breves y coloquiales como las presentes en las llamadas telefónicas.

Como resultado de esta estrategia, se obtuvo un dataframe vectorizado con TF-IDF que en las etapas siguientes fue aprovechado por los diferentes diseños de entrenamiento, en las siguientes imágenes se observa como una muestra de 10 documentos explícitamente sus vectores simplificados con sólo las 20 características más frecuentes, para inspeccionar de forma manual cómo están representadas las palabras clave en cada variante del texto definidas anteriormente [ver Figura 23, Figura 24, Figura 25 y Figura 26]

Ejemplos TF-IDF para `Text` — shape = (10, 20):

	bueno	con	de	el	en	es	gracias	la	me	no	para	por	pues	que	se	si	sí	te	un	ya
3648	0.000	0.000	0.422	0.450	0.000	0.545	0.0	0.000	0.000	0.567	0.000	0.000	0.000	0.0	0.000	0.00	0.000	0.0	0.000	0.000
819	0.000	0.000	0.100	0.000	0.119	0.000	0.0	0.000	0.135	0.942	0.144	0.000	0.166	0.0	0.000	0.00	0.000	0.0	0.000	0.149
9012	0.000	0.000	0.000	0.000	0.000	0.000	1.0	0.000	0.000	0.000	0.000	0.000	0.000	0.0	0.000	0.00	0.000	0.0	0.000	0.000
8024	0.000	0.000	0.000	0.000	0.000	0.000	0.0	0.000	0.000	0.000	0.000	0.000	0.000	0.0	0.000	0.00	0.000	0.0	0.000	0.000
7314	0.000	0.487	0.771	0.000	0.000	0.000	0.0	0.411	0.000	0.000	0.000	0.000	0.000	0.0	0.000	0.00	0.000	0.0	0.000	0.000
4572	0.801	0.000	0.000	0.598	0.000	0.000	0.0	0.000	0.000	0.000	0.000	0.000	0.000	0.0	0.000	0.00	0.000	0.0	0.000	0.000
3358	0.000	0.000	0.000	0.000	0.000	0.000	0.0	0.000	0.000	0.000	0.000	0.000	0.000	0.0	0.000	0.00	0.000	0.0	0.000	0.000
17870	0.000	0.462	0.488	0.000	0.000	0.000	0.0	0.520	0.000	0.492	0.000	0.000	0.000	0.0	0.000	0.19	0.000	0.0	0.000	0.000
2848	0.000	0.000	0.000	0.000	0.000	0.000	0.0	0.000	0.000	0.000	0.000	0.501	0.000	0.0	0.000	0.00	0.865	0.0	0.000	0.000
19349	0.000	0.000	0.106	0.113	0.000	0.000	0.0	0.000	0.286	0.854	0.153	0.000	0.000	0.0	0.172	0.00	0.000	0.0	0.334	0.000

Figura 23. Ejemplo TF-IDF para el texto original

Ejemplos TF-IDF para `Normalized_Text` — shape = (10, 20):

	bueno	con	de	el	en	es	gracias	la	me	no	para	por	pues	que	se	si	sí	te	un	ya
3648	0.000	0.000	0.422	0.450	0.000	0.545	0.0	0.000	0.000	0.567	0.000	0.000	0.000	0.0	0.000	0.00	0.000	0.0	0.000	0.000
819	0.000	0.000	0.100	0.000	0.119	0.000	0.0	0.000	0.135	0.942	0.144	0.000	0.166	0.0	0.000	0.00	0.000	0.0	0.000	0.149
9012	0.000	0.000	0.000	0.000	0.000	0.000	1.0	0.000	0.000	0.000	0.000	0.000	0.000	0.0	0.000	0.00	0.000	0.0	0.000	0.000
8024	0.000	0.000	0.000	0.000	0.000	0.000	0.0	0.000	0.000	0.000	0.000	0.000	0.000	0.0	0.000	0.00	0.000	0.0	0.000	0.000
7314	0.000	0.487	0.771	0.000	0.000	0.000	0.0	0.411	0.000	0.000	0.000	0.000	0.000	0.0	0.000	0.00	0.000	0.0	0.000	0.000
4572	0.801	0.000	0.000	0.598	0.000	0.000	0.0	0.000	0.000	0.000	0.000	0.000	0.000	0.0	0.000	0.00	0.000	0.0	0.000	0.000
3358	0.000	0.000	0.000	0.000	0.000	0.000	0.0	0.000	0.000	0.000	0.000	0.000	0.000	0.0	0.000	0.00	0.000	0.0	0.000	0.000
17870	0.000	0.462	0.488	0.000	0.000	0.000	0.0	0.520	0.000	0.492	0.000	0.000	0.000	0.0	0.000	0.19	0.000	0.0	0.000	0.000
2848	0.000	0.000	0.000	0.000	0.000	0.000	0.0	0.000	0.000	0.000	0.000	0.501	0.000	0.0	0.000	0.00	0.865	0.0	0.000	0.000
19349	0.000	0.000	0.106	0.113	0.000	0.000	0.0	0.000	0.286	0.854	0.153	0.000	0.000	0.0	0.172	0.00	0.000	0.0	0.334	0.000

Figura 24. Ejemplo TF-IDF para el texto normalizado

Ejemplos TF-IDF para `Lemmatized_Text` — shape = (10, 20):

	bueno	con	de	el	en	gracias	la	le	me	no	para	por	pues	que	se	si	sí	te	un	ya
3648	0.000	0.000	0.510	0.544	0.000	0.0	0.000	0.0	0.000	0.667	0.000	0.000	0.000	0.0	0.000	0.000	0.000	0.0	0.000	0.000
819	0.000	0.000	0.102	0.000	0.122	0.0	0.105	0.0	0.135	0.934	0.147	0.000	0.169	0.0	0.000	0.000	0.000	0.0	0.000	0.152
9012	0.000	0.000	0.000	0.000	0.000	1.0	0.000	0.0	0.000	0.000	0.000	0.000	0.000	0.0	0.000	0.000	0.000	0.0	0.000	0.000
8024	0.000	0.000	0.000	0.000	0.000	0.0	0.000	0.0	0.000	0.000	0.000	0.000	0.000	0.0	0.000	0.000	0.000	0.0	0.000	0.000
7314	0.000	0.489	0.775	0.000	0.000	0.0	0.400	0.0	0.000	0.000	0.000	0.000	0.000	0.0	0.000	0.000	0.000	0.0	0.000	0.000
4572	0.801	0.000	0.000	0.598	0.000	0.0	0.000	0.0	0.000	0.000	0.000	0.000	0.000	0.0	0.000	0.000	0.000	0.0	0.000	0.000
3358	0.000	0.000	0.000	0.000	0.000	0.0	0.000	0.0	0.000	0.000	0.000	0.000	0.000	0.0	0.000	0.000	0.000	0.0	0.000	0.000
17870	0.000	0.469	0.496	0.000	0.000	0.0	0.511	0.0	0.000	0.486	0.000	0.000	0.000	0.0	0.000	0.193	0.000	0.0	0.000	0.000
2848	0.000	0.000	0.000	0.000	0.000	0.0	0.000	0.0	0.000	0.000	0.000	0.501	0.000	0.0	0.000	0.000	0.865	0.0	0.000	0.000
19349	0.000	0.000	0.108	0.115	0.000	0.0	0.000	0.0	0.287	0.849	0.156	0.000	0.000	0.0	0.176	0.000	0.000	0.0	0.341	0.000

Figura 25. Ejemplo TF-IDF para el texto lematizado

Ejemplos TF-IDF para `Stemmed_Text` — shape = (10, 20):

	bueno	con	de	día	el	en	es	la	lo	me	no	para	por	pue	que	si	sí	te	un	ya
3648	0.000	0.000	0.426	0.0	0.454	0.000	0.55	0.000	0.0	0.000	0.557	0.000	0.000	0.000	0.0	0.000	0.000	0.0	0.000	0.000
819	0.000	0.000	0.102	0.0	0.000	0.122	0.00	0.105	0.0	0.135	0.934	0.147	0.000	0.169	0.0	0.000	0.000	0.0	0.000	0.152
9012	0.000	0.000	0.000	0.0	0.000	0.000	0.00	0.000	0.0	0.000	0.000	0.000	0.000	0.000	0.0	0.000	0.000	0.0	0.000	0.000
8024	0.000	0.000	0.000	0.0	0.000	0.000	0.00	0.000	0.0	0.000	0.000	0.000	0.000	0.000	0.0	0.000	0.000	0.0	0.000	0.000
7314	0.000	0.489	0.775	0.0	0.000	0.000	0.00	0.400	0.0	0.000	0.000	0.000	0.000	0.000	0.0	0.000	0.000	0.0	0.000	0.000
4572	0.768	0.000	0.000	0.0	0.641	0.000	0.00	0.000	0.0	0.000	0.000	0.000	0.000	0.000	0.0	0.000	0.000	0.0	0.000	0.000
3358	0.000	0.000	0.000	0.0	0.000	0.000	0.00	0.000	0.0	0.000	0.000	0.000	0.000	0.000	0.0	0.000	0.000	0.0	0.000	0.000
17870	0.000	0.469	0.496	0.0	0.000	0.000	0.00	0.511	0.0	0.000	0.486	0.000	0.000	0.000	0.0	0.193	0.000	0.0	0.000	0.000
2848	0.000	0.000	0.000	0.0	0.000	0.000	0.00	0.000	0.0	0.000	0.000	0.000	0.501	0.000	0.0	0.000	0.865	0.0	0.000	0.000
19349	0.000	0.000	0.110	0.0	0.117	0.000	0.00	0.000	0.0	0.291	0.863	0.158	0.000	0.000	0.0	0.000	0.000	0.0	0.347	0.000

Figura 26. Ejemplo TF-IDF para el texto con stemming

Con las columnas textuales de mayor señal discriminativa ya definidas y con las estrategias de balanceo de clases validadas empíricamente, se procedió a definir la columna Text como aquella que por temas de puntuación y contexto semántico seria la elegida para el proceso de los experimentos, toda vez, que al revisar el desempeño de las otras 3 versiones sus métricas finalmente no eran muy distantes, en los cuales se combinaron múltiples arquitecturas de aprendizaje profundo, configuraciones de entrenamiento y técnicas de validación. El objetivo fue determinar, bajo condiciones comparables y controladas, cuál combinación ofrecía el mejor desempeño en la clasificación multiclase de sentimientos, optimizando así la capacidad del modelo para interpretar y categorizar emociones expresadas en lenguaje natural durante interacciones telefónicas.

7.3. Descripción de los experimentos realizados

Partiendo de las conclusiones del capítulo 7.2 que identificó la representación textual más discriminativa y estableció la partición estratificada del corpus, en esta sección se describen las configuraciones experimentales utilizadas para entrenar los modelos.

7.3.1. Primer experimento

En este primer experimento denominado Entrenamiento de múltiples modelos de aprendizaje profundo se entrenaron y compararon cinco arquitecturas: un modelo *TextCNN* implementado en *PyTorch* con *embeddings* de 128 dimensiones y varias convoluciones de tamaños 3, 4 y 5 seguidas de *max-pool*; redes bidireccionales *Bi-LSTM* y *Bi-GRU*, ambas desarrolladas en *Keras* con 64 unidades por dirección y *embeddings* de 128 dimensiones; un modelo híbrido *CNN-LSTM* que combina sobre la salida de una capa convolucional local una capa *LSTM* con regularización por *dropout*; y un perceptrón multicapa (MLP) cuya entrada es la matriz *TF-IDF* de hasta 5.000 uni- y bigramas, dotado de una capa densa intermedia de 128 unidades y d*ropout* antes de la salida *softmax*.

Cada arquitectura se entrenó utilizando la función de pérdida de entropía cruzada y el optimizador Adam. Para evitar el sobreajuste y asegurar un entrenamiento eficiente, se empleó la técnica de *Early Stopping* con un parámetro de paciencia igual a dos, bajo la métrica de val_loss, lo que permitió detener automáticamente el entrenamiento cuando la pérdida de validación dejaba de mejorar. Aunque el número máximo de épocas se fijó en 20, la cantidad efectiva de épocas se ajustó dinámicamente según el desempeño de cada modelo. Esta técnica garantizó que cada red entrenara solo durante el número necesario de iteraciones para converger.

El conjunto de entrenamiento, previamente particionado de forma estratificada, fue sometido a cuatro estrategias de balanceo de clases (SMOTE, SMOTEENN, SMOTETomek y submuestreo aleatorio) y, en el caso de las secuencias, se aplicó un *oversampler* para garantizar la uniformidad de la estructura de los datos que recibía cada red. Cada combinación de arquitectura y método de muestreo se evaluó en el conjunto de prueba mediante las métricas de precisión, F1-macro, recall-macro y ROC-AUC macro para identificar aquellas configuraciones que ofrecieran el mejor compromiso entre exactitud global y sensibilidad hacia las clases minoritarias.

7.3.2. Segundo experimento

En este segundo experimento denominado Fine-Tuning Con Distilbeto, el entrenamiento del modelo basado en transformadores se llevó a cabo utilizando la arquitectura *DistilBERT* preentrenada para español, específicamente el modelo *dccuchile/distilbert-base-spanish-uncased*, el cual fue adaptado a una tarea de clasificación multiclase. La implementación se realizó con *PyTorch*, integrando la interfaz de *Hugging Face Transformers* para la carga y gestión del modelo y el *tokenizador*.

Se configuraron los parámetros fundamentales del entrenamiento: una longitud máxima de secuencia de 100 *tokens*, un *batch* de entrenamiento de 16 instancias y uno de validación de 32. El número máximo de épocas fue de 20, sujeto a la técnica de *early stopping* basada en la métrica de F1 macro sobre el conjunto de validación. Esto permitió detener el proceso de forma anticipada cuando ya no se observaban mejoras, garantizando así eficiencia y prevención del sobreajuste. El modelo fue cargado utilizando el método *from_pretrained*, especificando el número de clases (*NUM_CLASSES*) y permitiendo la adaptación del clasificador final mediante el parámetro *ignore_mismatched_sizes = True*, lo que facilitó su reutilización en una tarea con una cantidad de clases distinta al modelo base.

El *dataset* fue *tokenizado* previamente mediante el *AutoTokenizer* correspondiente, aplicando truncamiento y *padding* fijo hasta la longitud máxima permitida. La clase *SentimentDataset* fue definida para encapsular los tensores requeridos por el modelo: *input_ids, attention_mask y labels*, asegurando que todos ellos se desplazaran automáticamente al dispositivo de cómputo adecuado (*cuda* o *cpu*).

Posteriormente, se construyeron los objetos *DataLoader* para entrenamiento y validación, con los parámetros de *batch* definidos y *shuffle = True* activado para los datos de entrenamiento. Se utilizó el optimizador *AdamW*, ampliamente recomendado para modelos de *Transformers*, con una tasa de aprendizaje ajustada a *2e-5* para estabilizar la actualización de pesos en un escenario de *fine-tuning*.

7.3.3. Tercer experimento

En este tercer experimento, titulado Estrategia focal para clases minoritarias con DistilBETO, se buscó potenciar la detección de la clase "*Positivo*" mediante una serie de intervenciones diseñadas para contrarrestar el sesgo hacia la clase mayoritaria "*Neutral*". Para ello, se implementó un *oversampling* dirigido que replicó únicamente las instancias etiquetadas como positivas hasta igualar su cantidad a la clase neutral. A fin de aumentar la diversidad del conjunto, estas réplicas se sometieron a una técnica básica de *data*

augmentation, añadiendo un sufijo distintivo ("–*aug*") a las transcripciones positivas, con lo que se introdujo variación sintáctica sin alterar su significado esencial.

El modelo preentrenado DistilBETO se acondicionó para clasificación de tres etiquetas mediante la interfaz *AutoModelForSequenceClassification* de *Hugging Face*, conservando su capacidad de procesamiento contextual profundo en español. La entrada textual se *tokenizó* en lotes, aplicando truncamiento y *padding* para longitud fija, y luego se transformó en tensores adecuados para GPU o CPU. La función de pérdida central fue *Focal Loss*, configurada con un factor $\gamma = 2$ para enfatizar los ejemplos difíciles y un vector de pesos α que duplicó la importancia de la clase "*Positivo*" respecto a las demás. Este esquema permitió que durante el *fine-tuning* el modelo priorizara la minimización de errores en la clase minoritaria sin descuidar el rendimiento global.

El ajuste fino se realizó con un máximo de 20 épocas utilizando el optimizador *AdamW* con una tasa de aprendizaje de 2×10^{-5}. Para evitar el sobreajuste y garantizar un entrenamiento eficiente, se implementó la técnica de Early Stopping con un parámetro de paciencia igual a dos, deteniendo el proceso automáticamente cuando el F1-score macro de validación dejaba de mejorar. Cada época comprendió un ciclo de entrenamiento —activando el modo *"train()"* y calculando la pérdida de cada *batch* con retropropagación— seguido de una fase de validación —modo *"eval()"*, desactivación de gradientes y recopilación de predicciones— para monitorear métricas clave como precisión, *recall* y F1-score por clase. De esta manera, se evaluó cómo la combinación de *oversampling* selectivo, aumento de datos ligero y *Focal Loss* mejoraba la capacidad del modelo para reconocer con mayor frecuencia las instancias positivas difíciles, reduciendo al mismo tiempo la tendencia a etiquetar erróneamente ejemplos positivos como neutrales.

7.3.4. Cuarto experimento

El cuarto experimento, denominado Entrenamiento con validación cruzada 5-Fold, buscó medir con mayor rigor la capacidad de generalización de las cinco arquitecturas de aprendizaje profundo (TextCNN, Bi-LSTM, Bi-GRU, CNN-LSTM y MLP sobre TF-IDF). Partimos de la hipótesis de que un único particionado puede ocultar variaciones de rendimiento, especialmente con clases desbalanceadas. Por ello, empleamos *StratifiedKFold* para generar cinco conjuntos de entrenamiento y validación que preservaran la proporción de cada etiqueta. En cada ciclo se repitió todo el pipeline: preprocesamiento del texto, vectorización en paralelo para secuencias y TF-IDF, entrenamiento de los modelos y evaluación en el conjunto de validación.

Para las redes basadas en secuencias —TextCNN codificado en *PyTorch* y las arquitecturas recurrentes y mixtas en *Keras*— los textos se *tokenizaron* y *paddearon* a una

longitud fija. TextCNN se entrenó con *embeddings* de 128 dimensiones, tres tamaños de *Kernel* (3, 4 y 5) y una capa de *max-pooling*, entrenado con la función *CrossEntropyLoss* y el optimizador *Adam* a una tasa de aprendizaje de $2 \cdot 10^{-4}$, su entrenamiento se realizó con *Early Stopping*, permitiendo hasta 20 épocas, pero deteniéndose anticipadamente si la pérdida de validación no mejoraba durante dos iteraciones consecutivas bajo el parámetro de paciencia igual a dos en función de la métrica validadora val_loss. Bi-LSTM y Bi-GRU se configuraron con 64 unidades por dirección y capas de *embedding* de 128, mientras que el modelo híbrido aplicó una convolución *1D* seguida de una capa LSTM y regularización por *Dropout*. El MLP-TF-IDF recibió como entrada el vector TF-IDF de hasta 5.000 unigramas y bigramas, procesado por una capa densa de 128 unidades y *Dropout* antes de la *softmax* final. Cada uno de estos cuatro modelos de *Keras* se entrenó durante tres épocas usando *sparse_categorical_crossentropy* y *Adam*.

7.3.5. Quinto experimento

Para mitigar el sesgo de clase que persistía incluso después del *oversampling* clásico, se diseñó este quinto experimento denominado **Back-translation & Focal Oversample**, que combina la generación de datos sintéticos por *back-translation* y el uso de una función de pérdida focal ponderada. Primero, los textos de las clases "*Negativo*" y "*Positivo*" se tradujeron al inglés y posteriormente de vuelta al español mediante los modelos *MarianMT* (*es→en* y *en→es*), utilizados únicamente para la creación de nuevos ejemplos. Este procedimiento generó reformulaciones semánticamente equivalentes, pero con variaciones sintácticas, enriqueciendo así el conjunto de entrenamiento. Los ejemplos resultantes se validaron automáticamente y se integraron al corpus original, igualando el número de muestras en ambas clases minoritarias.

A continuación, el corpus extendido se procesó de forma paralela en dos representaciones: secuencias *tokenizadas* mediante un vocabulario restringido y *padding* hasta una longitud fija, y vectores TF-IDF de hasta 5.000 uni- y bigramas. Con el fin de enfatizar el aprendizaje sobre las clases más escasas, se calcularon pesos inversos a la frecuencia de cada etiqueta y se incorporaron a una versión multiclase de *Focal Loss* implementada en *TensorFlow* con *Keras backend*. Esta pérdida penaliza de manera más intensa los errores en las clases minoritarias, modulada por un parámetro α que dobla el peso de la clase "*Positivo*" y un factor $\gamma=2$ que amplifica el enfoque sobre ejemplos difíciles.

Se entrenaron dos modelos de forma independiente: un Bi-LSTM bidireccional con *embedding* de 300 dimensiones y 64 unidades por dirección, alimentado con las secuencias *tokenizadas*, y un perceptrón multicapa con dos capas densas y *dropout*, alimentado con los vectores TF-IDF. Ambos emplearon el optimizador *Adam*, validación interna para ajuste de hiperparámetros, Por su parte los modelos Keras implementaron *early stopping* para evitar

sobreajuste, ejecutando hasta 20 épocas con un parámetro de paciencia igual a dos bajo la métrica validadora por defecto val_loss.

Finalmente, las salidas *softmax* de los dos modelos se combinaron mediante un ensamble por promedio de probabilidades, con el propósito de aprovechar la sensibilidad contextual de la Bi-LSTM y la estabilidad estadística del MLP-TF-IDF. Para calibrar las probabilidades resultantes y mejorar su interpretabilidad, se ajustó un regresor logístico multiclase (*Platt Scaling*) sobre una partición del conjunto de prueba. Esta estrategia integró variación sintáctica, enfoque focal en clases minoritarias y calibración probabilística, logrando un sistema más equilibrado y confiable para la detección de sentimientos explícitos y sutiles.

8. EVALUACIÓN DEL DESEMPEÑO DEL MODELO DE CLASIFICACIÓN DE SENTIMIENTOS

En este apartado se describe la quinta fase del modelo CRISP-DM, denominada *Evaluación*, cuyo propósito es validar la efectividad del modelo en un entorno controlado mediante la aplicación de métricas específicas de desempeño propias del análisis de sentimientos. Esta fase permite determinar si el modelo cumple con los criterios de calidad establecidos y si está listo para su implementación. En este contexto, se desarrolla el cuarto objetivo específico de este proyecto aplicado.

8.1. Resultados obtenidos por experimentos

Los modelos entrenados para la tarea de clasificación de sentimientos fueron evaluados mediante métricas estándar: precisión (*precision*), exactitud (*accuracy*), recuperación (*recall*), pérdida *(loss)* y F1-score [48]. Asimismo, los modelos fueron evaluados utilizando visualizaciones, como la matriz de confusión, que permite ver el desempeño del modelo mostrando la cantidad de ejemplos correcta o incorrectamente clasificados por clase y así poder detectar patrones de confusión entre estas.

Estas métricas fueron calculadas utilizando funciones de *scikit-learn*, tanto durante el entrenamiento (validación por época) como al final del proceso de inferencia sobre el conjunto de validación. En los apartados siguientes se presentan los resultados detallados por modelo, así como un análisis comparativo que permitió seleccionar la mejor configuración para el problema abordado.

8.2. Resultados por experimento

8.2.1. Análisis de métricas primer experimento

Los resultados obtenidos del experimento *Entrenamiento múltiple modelos Deep Learning* nos permitió realizar una comparación entre estrategias de muestreo confirma que cada técnica influye de manera directa en el rendimiento de las arquitecturas de aprendizaje profundo frente al desequilibrio de clases.

Los resultados muestran que la técnica *SMOTE* genera una mejora generalizada en los modelos, con especial impacto en *Bi-GRU*, que logra su mayor *recall-macro* (0,5737), y en *MLP TF–IDF*, que alcanza su mejor *accuracy* (0,8056) dentro de todas las técnicas. *TextCNN* también se beneficia significativamente, obteniendo un buen equilibrio entre

accuracy (0,8198) y *F1-macro* (0,5292). En general, esta estrategia permite que todos los modelos mantengan un desempeño estable y balanceado entre métricas.

SMOTEENN aplica un filtrado híbrido que permite al *MLP TF-IDF* alcanzar su mayor *recall-macro* (0,6419), lo que evidencia su sensibilidad a instancias minoritarias tras el sobremuestreo y limpieza híbrida; sin embargo, esto viene acompañado de una fuerte caída en su *accuracy* (0,5697) y F1-macro (0,4229), lo que podría sugerir una mayor tasa de falsos positivos. El modelo *CNN→LSTM* presenta su peor desempeño bajo esta estrategia (F1-macro: 0,3937), mostrando poca estabilidad ante datos filtrados agresivamente.

SMOTETomek ofrece un desempeño sólido y balanceado, las *RNN* mantienen un buen equilibrio: *TextCNN* alcanza su mayor F1-macro (0,5315) y el segundo mejor AUC (0,8135), mientras que *Bi-LSTM* logra su mejor F1-macro (0,5292), un *accuracy* de 0,7891 y un *recall* de 0,5675. *MLP TF-IDF* también muestra técnicas consistentes en esta configuración, con un *ROC-AUC* macro destacado (0,7645). Esta estrategia demuestra ser eficaz tanto para modelos secuenciales como basados en *TF-IDF*, logrando métricas balanceadas en todos los frentes.

La técnica de submuestreo (*UnderSampler*) conserva la estabilidad en modelos como *Bi-GRU* (*recall*: 0,5435) y CNN→LSTM (*recall*: 0,5271), aunque sus métricas tienden a ser más moderadas. *TextCNN* obtiene su mejor *accuracy* global (0,8283) bajo esta estrategia, con un F1-macro competitivo (0,5219), evidenciando su capacidad de adaptarse a esquemas conservadores. En contraste, MLP TF–IDF sufre una caída drástica en F1-macro (0,3349), lo que indica una pérdida significativa de precisión en contextos de muestreo agresivo.

En conjunto, los resultados permiten concluir que *SMOTE* y *SMOTETomek* son las estrategias de balanceo más consistentes en términos generales, favoreciendo modelos como *Bi-LSTM, Bi-GRU* y *TextCNN*. *SMOTEENN*, en cambio, potencia el *recall* del *MLP TF–IDF* a costa de exactitud, y el *UnderSampler* resulta útil para modelos robustos a la reducción de datos, como *TextCNN*. La selección de la técnica de muestreo debe hacerse en función del objetivo: mejorar *recall* para clases minoritarias o mantener precisión global del sistema. [ver Tabla 8]

Tabla 8. Resultados de validación de cada modelo por técnica de balanceo

Estrategia	Modelo	accuracy	f1_macro	recall_macro	roc_auc_macro
SMOTE	**TextCNN**	0,819824	0,529193	0,530458	0,804279
	Bi-LSTM	0,805369	0,513890	0,536323	0,747756
	Bi-GRU	0,787042	0,514162	0,573700	0,781536
	CNN→LSTM	0,778265	0,516780	0,541530	0,716910

Estrategia	Modelo	accuracy	f1_macro	recall_macro	roc_auc_macro
	MLP TF–IDF	0,805627	0,515700	0,552271	0,767318
SMOTEENN	**TextCNN**	0,811564	0,498872	0,496152	0,782568
	Bi-LSTM	0,803046	0,510326	0,521783	0,726188
	Bi-GRU	0,802530	0,510110	0,514912	0,738449
	CNN→LSTM	0,835570	0,393782	0,449549	0,657717
	MLP TF–IDF	0,569695	0,422895	0,641968	0,786811
SMOTETomek	**TextCNN**	0,831699	0,531533	0,528691	0,813509
	Bi-LSTM	0,789107	0,529190	0,567532	0,753109
	Bi-GRU	0,801239	0,514005	0,531916	0,736861
	CNN→LSTM	0,764326	0,499929	0,531874	0,682792
	MLP TF–IDF	0,804595	0,513387	0,548047	0,764548
UnderSampler	**TextCNN**	0,828343	0,521883	0,506824	0,800590
	Bi-LSTM	0,798141	0,496425	0,527401	0,716061
	Bi-GRU	0,803562	0,517141	0,543482	0,740827
	CNN→LSTM	0,793237	0,481620	0,527070	0,691528
	MLP TF–IDF	0,705989	0,334980	0,556068	0,687433

En la visualización del *heatmap* [ver Figura 27] se evidencia que *TextCNN* se consolidó como la arquitectura más consistente en términos de F1-macro, alcanzando los valores más altos en casi todas las estrategias de muestreo. Su mejor desempeño se registra con *SMOTETomek* (F1-macro: 0,5315), lo que resalta su robustez frente a variaciones en el balanceo de clases. La estrategia *SMOTEENN* mantiene resultados relativamente estables para *Bi-GRU* y *Bi-LSTM* (ambos con F1-macro de 0,51), pero afecta considerablemente al *CNN→LSTM*, cuyo F1-macro cae abruptamente hasta 0,39, siendo el punto más bajo entre todos los modelos y estrategias. Este comportamiento sugiere una baja tolerancia de las redes híbridas a las limpiezas agresivas propias de *SMOTEENN*. Por otro lado, el modelo *MLP TF–IDF* muestra una alta sensibilidad al tipo de muestreo. Aunque logra un F1-macro competitivo bajo *SMOTE* (0,52), su rendimiento cae drásticamente hasta 0,33 con *UnderSampler*, lo que indica una pérdida significativa de capacidad para generalizar correctamente ante clases desbalanceadas cuando se reduce el tamaño del conjunto de entrenamiento.

En términos generales, las estrategias *SMOTE* y *SMOTETomek* emergen como las más efectivas para sostener un rendimiento elevado y estable, especialmente en arquitecturas como *TextCNN* y *Bi-LSTM*, que muestran mejoras consistentes en estas condiciones. La visualización confirma que la combinación *TextCNN + SMOTETomek* ofrece el mejor compromiso entre robustez y rendimiento. En contraste, modelos como *CNN→LSTM* y *MLP TF–IDF* requieren un ajuste más cuidadoso del muestreo, ya que son más susceptibles a desequilibrios y modificaciones en la distribución de clases.

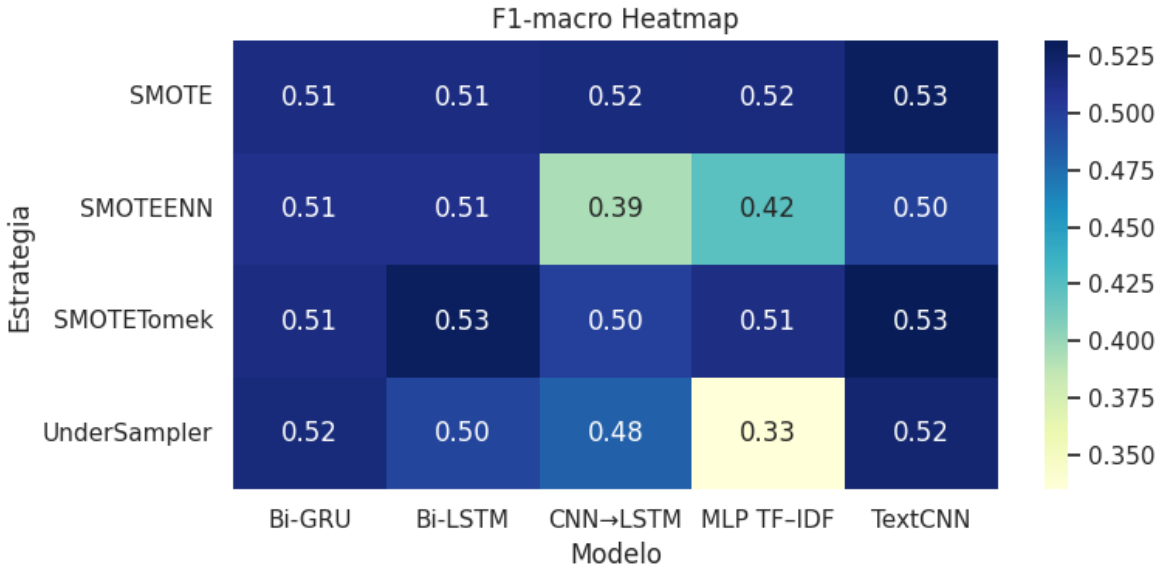


Figura 27. Mapa de calor de los modelos

En concordancia, con los resultados obtenidos se llevó a cabo un proceso de inferencia para evaluar la calidad de las predicciones obteniendo los siguientes resultados [ver Tabla 9].

Tabla 9. Prueba de inferencia por modelo

text	**TextCNN**	**Bi-LSTM**	**Bi-GRU**	**CNN-LSTM híbrido**	**MLP TF–IDF**
Me encanta, pago hoy mismo el semestre	Neutral	Neutral	Neutral	Neutral	Neutral
No me gusta, el programa académico es muy malo	Neutral	Negativo	Negativo	Neutral	Negativo
Entiendo, lo consultaré con mis papas...	Neutral	Neutral	Neutral	Negativo	Negativo
deme su numero y si algo le llamo despues	Neutral	Neutral	Neutral	Neutral	Neutral
Gracias pero no estoy interesado	Negativo	Negativo	Negativo	Negativo	Negativo
cual es el horario de atencion	Neutral	Neutral	Positivo	Neutral	Neutral

Los modelos divergieron de manera notable al clasificar estos seis ejemplos, revelando diferencias importantes en su sensibilidad ante señales sutiles de polaridad. **TextCNN** mostró un comportamiento fuertemente sesgado hacia la clase "*Neutral*", asignando esta etiqueta en cinco de las seis frases y reconociendo "*Negativo*" solo cuando el

rechazo fue explícito ("*Gracias, pero no estoy interesado*"). Este comportamiento sugiere una marcada tendencia del modelo a optar por decisiones conservadoras, salvo en contextos donde el lenguaje negativo es explícito.

Por su parte, **Bi-GRU** compartió la misma clasificación que **Bi-LSTM** salvo en la frase final ("*cuál es el horario de atención*"), una decisión llamativa considerando que se trata de una consulta informativa. Esto podría indicar una mayor sensibilidad del modelo a ciertos términos o estructuras que asocia con intención positiva, aunque en este caso la predicción parece más una excepción que una regla. El modelo **CNN→LSTM híbrido** también adoptó una postura conservadora, salvo en dos casos: etiquetó como *Negativa* la frase *"Entiendo, lo consultaré con mis papás...",* lo que sugiere una interpretación más estricta de las expresiones condicionales o evasivas, y acertó en la clasificación negativa de la frase de rechazo explícito. Aun así, en general se mantuvo dentro del rango neutral predominante.

En contraste, el **MLP sobre TF–IDF** fue el modelo más variado en sus predicciones. Acierta en los ejemplos con contenido claramente negativo ("*No me gusta…*" y "*Gracias, pero no estoy interesado*"), donde otros modelos mostraron indecisión. Sin embargo, no identificó ningún caso como *Positivo*, lo que refleja una mayor propensión a detectar polaridad negativa que positiva, y una posible sensibilidad a palabras con connotación desfavorable.

En conjunto, este ejercicio evidencia cómo los modelos basados en redes secuenciales tienden a neutralizar las decisiones en presencia de ambigüedad, posiblemente como resultado de un balanceo de clases aplicado durante el entrenamiento. Mientras tanto, el *MLP con TF–IDF* y el *Bi-GRU* muestran una mayor propensión a responder a señales específicas dentro del texto, lo que puede aumentar su sensibilidad, pero también su variabilidad. La elección del modelo, por tanto, debería considerar el grado de matiz emocional esperado en los datos y la tolerancia al riesgo de falsas interpretaciones en contextos sutiles.

8.2.2. Análisis de métricas segundo experimento

En esta ocasión el experimento con *DistilBETO fine-tuned* logró un *accuracy* global del 86 %, manteniendo un rendimiento destacado en la clase mayoritaria, pero evidenciando dificultades importantes en las clases minoritarias. En la clase "*Neutral*", que representa la gran mayoría de los ejemplos, el modelo alcanza una precisión del 89 % y un *recall* del 96 %, con un F1-score de 0,92, lo que confirma su capacidad para identificar de forma muy confiable los casos neutros.

Sin embargo, este dominio se ve contrastado por el débil desempeño en "*Negativo*" y "*Positivo*". Para la clase "*Negativo*", la precisión se ubica en 0,58, pero el *recall* cae al 34 %, reflejando que casi 7 de cada 10 ejemplos negativos son mal clasificados, en su mayoría como neutros. El F1-score resultante es de solo 0,43, indicando una alta pérdida informativa. La clase "*Positivo*" enfrenta aún más dificultades: con una precisión del 50 %, pero un *recall* de apenas 27 %, su F1-score cae a 0,35, lo que significa que más de tres cuartas partes de los ejemplos positivos reales no son reconocidos como tales.

El promedio macro de las métricas —precisión 0,66, recall 0,52, F1-score 0,57— resume esta brecha de desempeño entre clases. A pesar del sólido resultado global y del rendimiento casi perfecto en la clase dominante, el modelo muestra una clara incapacidad para capturar patrones representativos de las clases minoritarias. Esto pone en evidencia la necesidad de estrategias adicionales como reajuste de umbrales de decisión, entrenamiento con pérdidas focales ponderadas o incorporación de datos sintéticos mediante técnicas como *oversampling* dirigido o *back-translation*, con el objetivo de reducir la asimetría y mejorar la sensibilidad en "*Negativo*" y "*Positivo*" sin sacrificar la estabilidad lograda en "*Neutral*".

La matriz de confusión evidencia que el modelo está altamente optimizado para la clase "Neutral", pero lo hace a costa de una pérdida considerable de precisión en "Negativo" y "Positivo". Las clases minoritarias son absorbidas por la dominante, lo que indica un desequilibrio típico en tareas multiclase con distribución sesgada [ver Figura 28]

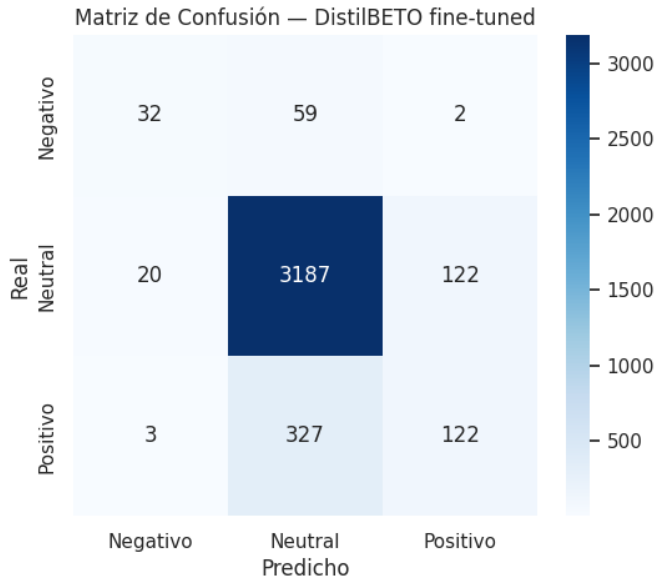


Figura 28. Matriz de confusión modelo DistilBETO fine-tuned

En concordancia, con los resultados obtenidos se llevó a cabo un proceso de inferencia para evaluar la calidad de las predicciones obteniendo los siguientes resultados. [ver Tabla 10]

Tabla 10. Resultados de inferencia modelo DistilBETO fine-tuned

text	**DistilBETO**
Me encanta, pago hoy mismo el semestre	Positivo
No me gusta, el programa académico es muy malo	Negativo
Entiendo, lo consultaré con mis papas...	Positivo
deme su numero y si algo le llamo despues	Neutral
Gracias pero no estoy interesado	Negativo
cual es el horario de atencion	Neutral

En las inferencias de ejemplo [ver Tabla 10], el modelo *DistilBETO fine-tuned* clasifica correctamente frases con polaridad evidente: "Me encanta, pago hoy mismo el semestre" es identificada como *Positivo*, mientras que "No me gusta, el programa académico es muy malo" se clasifica como *Negativo*, en línea con la intuición humana y confirmando que el modelo es capaz de detectar expresiones afectivas explícitas.

Por otro lado, frases como "*deme su número y si algo le llamo después*", y *"¿cuál es el horario de atención?*" son clasificadas como *Neutral*, lo cual refleja una tendencia del modelo a agrupar en esta clase aquellos casos donde el lenguaje es más vago, formal o informativo. Esta inclinación también se observa en la frase "*Gracias, pero no estoy interesado*", que sí es correctamente etiquetada como *Negativo*, gracias a la presencia de una negación explícita que el modelo ha aprendido a reconocer tras el *fine-tuning.*

Este comportamiento es coherente con las métricas globales alcanzadas por el modelo: una alta precisión global (*accuracy = 0,8624*) y una robustez notable sobre la clase "*Neutral*", pero también una recuperación limitada de sentimientos minoritarios, reflejada en un *recall* macro de 0,5238 y un F1-macro de 0,5685. En conjunto, estos resultados indican que el ajuste fino realizado sobre DistilBETO ha mejorado sustancialmente su capacidad de adaptación al dominio de análisis de sentimientos, pero que persisten desafíos para lograr una detección más equilibrada, especialmente en casos donde la polaridad emocional es implícita o menos frecuente. Esto refuerza la necesidad de incorporar estrategias complementarias como reentrenamiento focalizado, balanceo de clases, o ajuste de umbrales de decisión por clase.

8.2.3. Análisis de métricas tercer experimento

Como punto de partida del experimento anterior, este fue denominado Estrategia focal clases minoritarias con DistilBeto, nos permitió ver que luego de completadas 5 épocas de entrenamiento, las siguientes métricas obtenidas [ver Tabla 11] el modelo alcanza una *accuracy* global de 0,82, con un *recall* y *F1-macro* de 0,58, lo que representa una mejora global en la sensibilidad comparado con configuraciones anteriores. Al desagregar por clase, la clase "*Neutral*" sigue siendo la más sólidamente identificada, con un *recall* del 89 % y un F1-score de 0,90, reflejando la robustez del modelo ante ejemplos mayoritarios. La clase "*Negativo*" alcanza un *recall* del 48 %, mejorando sustancialmente respecto a versiones anteriores, aunque su precisión se reduce a 0,46, lo que indica un mayor número de falsos positivos respecto a esta categoría. La clase "*Positivo*" muestra un *recall* de 0,37 y un F1 de 0,36, resultado de la combinación entre *oversample* dirigido y *Focal Loss*, que favorece la detección de instancias positivas aún a costa de reducir precisión [ver Tabla 11].

Tabla 11. Métricas por clase DistilBETO

Clase	Precision	Recall	F1-Score	Support
Negativo	0,46	0,48	0,47	93
Neutral	0,90	0,89	0,90	3.329
Positivo	0,35	0,37	0,36	452
Accuracy			0,82	3874
Macro avg	0,57	0,58	0,58	3874
Weighted avg	0,83	0,82	0,82	3874

Al aplicar un ajuste de umbral para la clase "*Positivo*", reduciéndolo de 0,50 a 0,30, el *recall* alcanza un valor de 0,471, lo que indica que el modelo logra detectar casi 5 de cada 10 ejemplos positivos reales. Esta mejora valida la hipótesis de que estrategias específicas de refuerzo para clases minoritarias —como el uso de *Focal Loss* ponderada y el *tuning* de umbrales— pueden aumentar significativamente la sensibilidad. Sin embargo, el resultado también sugiere la existencia de un *trade-off* en precisión, especialmente sobre las clases "*Negativo*" y "*Neutral*", evidenciando la necesidad de un ajuste fino que equilibre detección con exactitud al aplicar este tipo de estrategias en entornos multiclase desbalanceados.

En la Tabla 12 se puede observar el proceso de inferencia para los textos de prueba para las predicciones, de esta tabla podemos destacar que Los resultados muestran que el modelo no solo asigna etiquetas a cada ejemplo de prueba, sino que también expresa un grado explícito de confianza en su decisión a través de las probabilidades asociadas a cada clase.

Esta información permite evaluar no solo la predicción final, sino también la calibración interna del modelo.

En la primera oración —"*Me encanta, pago hoy mismo el semestre*"— el modelo predice *Positivo* con una probabilidad dominante de 0,77, y niveles muy bajos para las otras clases (*Negativo*: 0,001, *Neutral*: 0,23). Esto refleja una alta confianza en su decisión y una buena capacidad para reconocer expresiones de entusiasmo explícitas.

Para "*No me gusta, el programa académico es muy malo*", la predicción es *Negativo* con una probabilidad de 0,98, lo cual también evidencia una respuesta muy segura. Las probabilidades de *Neutral* (0,018) y *Positivo* (0,0005) son marginales, lo que indica que el modelo ha aprendido bien las señales lingüísticas asociadas a juicios negativos.

Tabla 12. Proceso de inferencia modelo DistilBETO fine-tuned

text	prediction	probabilities	prediction_readable	p_Negativo	p_Neutral	p_Positivo
Me encanta, pago hoy mismo el semestre	LABEL_2	[0,0015436802, 0,2318988, 0,7665575]	Positivo	0,001544	0,231899	0,766558
No me gusta, el programa académico es muy malo	LABEL_0	[0,9819062, 0,017551934, 0,00054190407]	Negativo	0,981906	0,017552	0,000542
Entiendo, lo consultaré con mis papas...	LABEL_1	[0,001489328, 0,4515225, 0,5469882]	Positivo	0,001489	0,451522	0,546988
deme su numero y si algo le llamo despues	LABEL_1	[0,0008977254, 0,98145884, 0,01764347]	Neutral	0,000898	0,981459	0,017643
Gracias pero no estoy interesado	LABEL_0	[0,99516547, 0,0037472127, 0,0010873149]	Negativo	0,995165	0,003747	0,001087
cual es el horario de atencion	LABEL_1	[0,0020613659, 0,9970963, 0,0008423473]	Neutral	0,002061	0,997096	0,000842

El caso de "*Entiendo, lo consultaré con mis papás…*", es especialmente interesante: aunque el tono puede parecer ambiguo, el modelo opta por *Positivo* con 0,547, superando a *Neutral* (0,4515) y dejando *Negativo* en apenas 0,0015. Esta elección sugiere que el modelo interpreta ciertas expresiones de disposición futura como una señal positiva, aunque

matizada. La proximidad entre *Positivo* y *Neutral* también indica que el modelo es sensible a este tipo de ambigüedad.

Con "*deme su número y si algo le llamo después*", el modelo predice con firmeza Neutral (0,9815), descartando Positivo (0,0176) y Negativo (0,0009). Aunque la frase podría tener una intención favorable implícita, la elección del modelo refleja prudencia, interpretando la estructura como una respuesta cortés sin carga emocional fuerte.

La frase *"Gracias, pero no estoy interesado"* es clasificada como *Negativo* con una altísima certeza (0,9952), lo que demuestra una clara comprensión del modelo ante mensajes de rechazo directo. Las probabilidades de las clases *Neutral* y *Positivo* son marginales (ambas por debajo de 0,004), lo que refuerza la confianza del modelo en su predicción.

Finalmente, *"¿cuál es el horario de atención?*" es clasificada como *Neutral* con una probabilidad bastante alta (0,997), mientras que las clases *Negativo* y *Positivo* quedan por debajo de 1 %. Este resultado muestra que el modelo discrimina con precisión las consultas informativas, manteniéndolas dentro de un marco afectivo neutro.

En conjunto, los ejemplos muestran un modelo bien calibrado y consistente: las clases predichas no solo son intuitivamente correctas, sino que las probabilidades asociadas reflejan un nivel claro de certeza, con márgenes amplios entre la clase seleccionada y las alternativas. Esto evidencia que *DistilBETO fine-tuned* no solo acierta, sino que lo hace con confianza estructurada, lo que es esencial para tareas sensibles donde el umbral de decisión puede ajustarse según el contexto de uso.

8.2.4. Análisis de métricas cuarto experimento

En esta ocasión, los resultados de la validación cruzada (*5-Fold estratificada*) muestran que los modelos *Bi-GRU y MLP TF–IDF* alcanzan las mejores precisiones medias: 0,8636 y 0,8652, respectivamente, con desviaciones estándar bajas (< 0,003), lo cual indica un comportamiento estable y reproducible en términos de *accuracy*. Sin embargo, al examinar métricas más sensibles al balance entre clases, como el F1-macro y el *recall*-macro, se observa un panorama distinto.

El modelo *TextCNN* se posiciona como el mejor en F1-macro (0,5155), seguido de cerca por *Bi-GRU* (0,4908) y *Bi-LST* (0,4785). Esto sugiere que, aunque no lideran en exactitud global, las arquitecturas recurrentes bidireccionales y el modelo MLP ofrecen un rendimiento más equilibrado entre precisión y sensibilidad en clases desbalanceadas. En contraste, el modelo CNN–LSTM híbrido presenta un F1 bajo (0,3119) y también el *recall*

más bajo de todos (0,3349), acompañado de la mayor desviación estándar en ROC-AUC (0,0942), lo cual indica inestabilidad y pobre generalización *interfold*.

En términos de *recall* macro, la *TextCNN* también lidera (0,4699), seguida por *Bi-LSTM* (0,4535) y *Bi-GRU* (0,4506). Estas cifras muestran que los modelos recurrentes tienen una mayor capacidad para recuperar correctamente las clases minoritarias, lo cual es clave en tareas con fuerte desbalance. Finalmente, el análisis del ROC-AUC macro ubica a *Bi-LSTM* (0,8344), MLP (0,8329) y TextCNN (0,8328) como los más competitivos en términos de discriminación entre clases, confirmando que su desempeño no solo es bueno en clasificación, sino también en calibración probabilística.

En conjunto, en la Tabla 13, los resultados sugieren que Bi-GRU y Bi-LSTM ofrecen el mejor compromiso entre rendimiento global y sensibilidad multiclase, mientras que TextCNN y MLP destacan por su precisión general y estabilidad. Por el contrario, el modelo CNN–LSTM híbrido muestra resultados sistemáticamente inferiores en todas las métricas evaluadas, por lo que requeriría ajustes significativos en arquitectura o estrategia de entrenamiento para resultar competitivo.

Tabla 13. Resultados de calidad de los 5 modelos

Modelo	**accuracy_mean**	**accuracy_std**	**f1_mean**	**f1_std**	**recall_mean**	**recall_std**	**roc_auc_mean**	**roc_auc_std**
TextCNN	0,860602	0,001698	0,515523	0,031339	0,469965	0,028400	0,832814	0,008605
Bi-LSTM	0,863132	0,002136	0,478479	0,042353	0,453480	0,035349	0,834394	0,012467
Bi-GRU	0,863596	0,002231	0,490757	0,018033	0,450646	0,018138	0,830402	0,010594
CNN-LSTM	0,858743	0,000607	0,311924	0,005678	0,334878	0,002498	0,669097	0,094185
MLP	0,865197	0,002656	0,478173	0,016373	0,437069	0,012777	0,832923	0,006612

Como se observa en la Tabla 14, los cinco modelos presentan una marcada tendencia a la sobre clasificación hacia la clase "*Neutral*", independientemente de la arquitectura empleada. Etiquetan la mayoría de las frases como neutrales, incluso en casos donde el contenido semántico sugiere una polaridad más clara. La única excepción significativa se presenta en la frase "Gracias, pero no estoy interesado", que es correctamente clasificada como "*Negativo*" por los modelos TextCNN, Bi-LSTM y Bi-GRU, mientras que CNN–LSTM y MLP continúan asignándole una etiqueta neutral. Y por otro lado la expresión *"Me*

encanta, pago hoy mismo el semestre" solo es clasificada como *"Positivo"* por el modelo TextCNN.

Tabla 14. Resultados de inferencia de los 5 modelos

text	TextCNN	Bi-LSTM	Bi-GRU	CNN-LSTM híbrido	MLP TF–IDF
Me encanta, pago hoy mismo el semestre	Positivo	Neutral	Neutral	Neutral	Neutral
No me gusta, el programa académico es muy malo	Neutral	Neutral	Neutral	Neutral	Neutral
Entiendo, lo consultaré con mis papas...	Neutral	Neutral	Neutral	Neutral	Neutral
deme su numero y si algo le llamo despues	Neutral	Neutral	Neutral	Neutral	Neutral
Gracias pero no estoy interesado	Negativo	Negativo	Negativo	Neutral	Neutral
cual es el horario de atencion	Neutral	Neutral	Neutral	Neutral	Neutral

Este comportamiento sugiere la existencia de un sesgo común hacia la clase mayoritaria, posiblemente derivado del desbalance presente en los datos de entrenamiento. La falta de sensibilidad ante frases con carga emocional, como aquellas con rechazo explícito o afirmaciones positivas, evidencia una limitación compartida en la capacidad de los modelos para discriminar entre polaridades. Esta observación es consistente con los resultados macro de las métricas obtenidas previamente: altos niveles de precisión general, pero pobre desempeño en el reconocimiento de clases minoritarias.

En conjunto, estos hallazgos refuerzan la necesidad de aplicar estrategias complementarias, como el reentrenamiento focalizado, el ajuste de umbrales de decisión o el uso de técnicas de generación de datos sintéticos, con el fin de mejorar la capacidad de detección de sentimientos no neutros en contextos más sutiles o ambiguos.

8.2.5. Análisis de métricas quinto experimento

Tras el entrenamiento del experimento *Back-Traslation & Focal Oversample* conjunto del Bi-LSTM y el MLP sobre TF–IDF, su ensamble por promedio de probabilidades, y la posterior calibración con *Platt Scaling*, se obtienen mejoras sustanciales en la calidad de las predicciones, especialmente en términos de balance multiclase. El *Log-Loss* final de 0,42 confirma que la calibración ha permitido ajustar las probabilidades emitidas por el modelo, acercándolas a una representación probabilística más confiable. El *accuracy* global se mantiene en 0,83, muy competitivo frente al mejor modelo individual, pero con ventajas adicionales en métricas de equidad y sensibilidad [ver Figura 29].

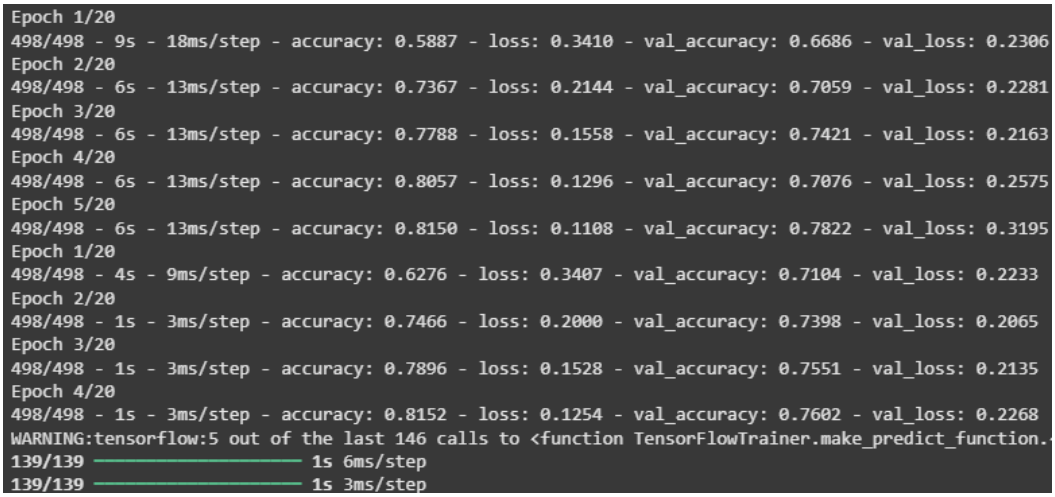


Figura 29. Épocas de entrenamiento

En particular, el *recall* de la clase "*Negativo*" alcanza el 45 %, y el de "*Positivo*" también llega al 49 %, lo que representa una mejora considerable frente a configuraciones previas, donde estos valores solían estar por debajo del 30 %. Las precisiones asociadas a estas clases (0,65 para Negativo y 0,73 para Positivo) demuestran que el modelo no solo detecta mejor las clases minoritarias, sino que lo hace sin un aumento drástico en falsos positivos. El desempeño sobre la clase "*Neutral*" se mantiene alto, con un *recall* del 95 % y un F1 de 0,90, lo que asegura que la mejora en las clases minoritarias no se da a expensas de la clase dominante [ver Tabla 15].

Tabla 15. Métricas en la calidad del entrenamiento

Ensamble + Platt Scaling				
Clase	**Precision**	**Recall**	**F1-Score**	**Support**
Negativo	0,65	0,45	0,53	93
Neutral	0,85	0,95	0,90	1.665
Positivo	0,73	0,49	0,59	452
Accuracy			0,83	2.210
Macro avg	0,74	0,63	0,67	2.210
Weighted avg	0,82	0,83	0,82	2.210
Con Threshold Tuning				
Clase	**Precision**	**Recall**	**F1-Score**	**Support**
Negativo	0,65	0,45	0,53	93
Neutral	0,86	0,94	0,90	1.665
Positivo	0,73	0,50	0,59	452
Accuracy			0,83	2.210

Macro avg	0,74	0,63	0,67	2.210
Weighted avg	0,82	0,83	0,82	2.210

El F1-macro resultante de 0,67, junto con un *macro-recall* de 0,63, supera con claridad los resultados obtenidos antes de aplicar calibración y ensamblado, validando que la combinación de arquitecturas permite mejorar la cobertura general sin comprometer la robustez. Tras aplicar *tuning* de umbral (reduciendo el punto de corte de la clase "*Positivo*" de 0,50 a 0,30), se observa una ligera mejora adicional en el *recall* positivo (0,49 → 0,50), manteniéndose estables las demás métricas.

En conjunto, estos resultados respaldan la hipótesis de que el uso combinado de técnicas como *Focal Loss, oversampling* selectivo, ensamblado de modelos complementarios y calibración probabilística puede corregir el sesgo hacia la clase mayoritaria, logrando un sistema de clasificación más equitativo y sensible en escenarios de sentimiento altamente desbalanceados.

En la Tabla 16 se puede observar el resultado de las inferencias bajo la hipótesis de que un umbral ajustado para la clase "*Positivo*" (≥ 0,40) permitiría aumentar la sensibilidad sin sacrificar la precisión general, los resultados de inferencia confirman dicha expectativa con matices importantes. La frase "*Me encanta, pago hoy mismo el semestre*" alcanza un 0,58 de probabilidad para *Positivo* y es correctamente clasificada como tal, reflejando una detección acertada de intención positiva. En contraste, *"Gracias, pero no estoy interesado"* obtiene una probabilidad de 0,81 para "Negativo", lo que valida la robustez del modelo ante frases con carga negativa explícita.

Otros casos muestran matices interesantes. La frase *"No me gusta, el programa académico es muy malo"* fue clasificada como *"Neutral"*, a pesar de que *"Negativo"* tiene una probabilidad considerable (0,34). Este resultado sugiere una cierta confusión entre rechazo implícito y descripciones críticas, posiblemente influida por la cercanía entre las probabilidades asignadas a "*Negativo*" (0,34) y "*Neutral*" (0,38). Una situación similar ocurre con *"Entiendo, lo consultaré con mis papás…"*, clasificada como "*Negativo*" al superar el umbral mínimo con 0,65, aunque con evidencias de ambigüedad en la interpretación emocional.

En frases de naturaleza informativa como *"deme su número…"* y *"¿cuál es el horario de atención?"*, el modelo las clasifica correctamente como "*Neutral*" (con probabilidades superiores al 50 %), lo que sugiere una adecuada detección del lenguaje sin carga afectiva.

En conjunto, las decisiones del modelo muestran coherencia con el ajuste de umbrales aplicado: los casos positivos con alta certeza son capturados correctamente, los neutros se mantienen estables, y las decisiones en zonas grises tienden a favorecer la clase con mayor margen probabilístico, reduciendo el riesgo de falsos positivos. Esto valida la hipótesis de que un ajuste fino de umbrales, junto con un sistema de inferencia calibrado, puede mejorar la recuperación de clases minoritarias sin comprometer la precisión general del sistema. No obstante, los resultados también evidencian la necesidad de explorar estrategias complementarias —como el refinamiento semántico, técnicas de interpretabilidad o mecanismos de reentrenamiento focalizado— para abordar con mayor eficacia los casos limítrofes entre polaridades sutiles.

Tabla 16. Inferencias bajo un umbral ajustado para la clase "Positivo"

text	Negativo (p)	Neutral (p)	Positivo (p)	predicted
Me encanta, pago hoy mismo el semestre	0.091844	0.326731	0.581425	Positivo
No me gusta, el programa académico es muy malo	0.344176	0.385063	0.270761	Neutral
Entiendo, lo consultaré con mis papas...	0.652089	0.149485	0.198426	Negativo
deme su numero y si algo le llamo despues	0.145914	0.518622	0.335464	Neutral
Gracias pero no estoy interesado	0.814930	0.068078	0.116992	Negativo
cual es el horario de atencion	0.186533	0.601007	0.212461	Neutral

Las matrices de confusión —tanto en conteo absoluto como normalizadas por fila— revelan de forma clara el comportamiento del modelo tras la aplicación completa del *pipeline* de ensamblado, calibración con *Platt Scaling* y ajuste de umbral. La clase "*Neutral*" se mantiene como la mejor identificada: el modelo logra reconocer correctamente el 94 % de sus casos, es decir, 1.573 de 1.665 ejemplos, con solo 21 confundidos como "*Negativo*" y 71 como "*Positivo*". Esta robustez en la clase mayoritaria sustenta la alta precisión general (*accuracy = 0,83*).

Para la clase "*Negativo*", el modelo logra recuperar correctamente el 43 % de los casos (45 de 93), pero aún confunde un 43 % como "*Neutral*". Solo 8 son erróneamente asignados a "*Positivo*". Esta distribución sugiere que, aunque la calibración y el *oversample* han mejorado su detección respecto a configuraciones previas, la frontera entre "*Negativo*" y "*Neutral*" sigue siendo estrecha, especialmente cuando el lenguaje es indirecto o ambiguo.

En la clase "*Positivo*", el modelo alcanza un *recall* del 43 %, con 195 ejemplos correctamente clasificados, pero aún 252 son confundidos como "*Neutral*", y solo 5 como

"*Negativo*". Esto confirma que el modelo ha mejorado su sensibilidad hacia expresiones positivas, aunque tiende a seguir favoreciendo la neutralidad en casos menos explícitos. Esto es consistente con inferencias previas, como en la frase "*Entiendo, lo consultaré con mis papás…*", donde pese a tener un valor bajo para *Positivo* ($p \approx 0{,}20$), el modelo decide por una clase negativa o neutral ante la falta de certeza [ver Figura 30].

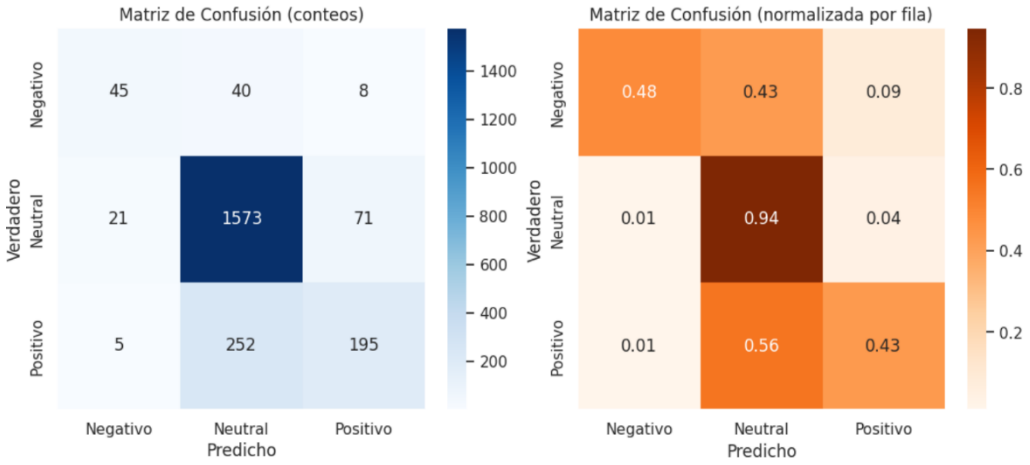


Figura 30. Matriz de confusión del modelo

El reporte de clasificación [ver Figura 31] respalda visualmente estos hallazgos: se observan F1-scores de 0,55, 0,89 y 0,54 para *Negativo*, *Neutral* y *Positivo* respectivamente, y un macro-F1 de 0,66, que mejora respecto a configuraciones sin calibración ni ensamblado. También se mantiene un *macro-recall* de 0,62, reflejando una recuperación más justa de las clases minoritarias.

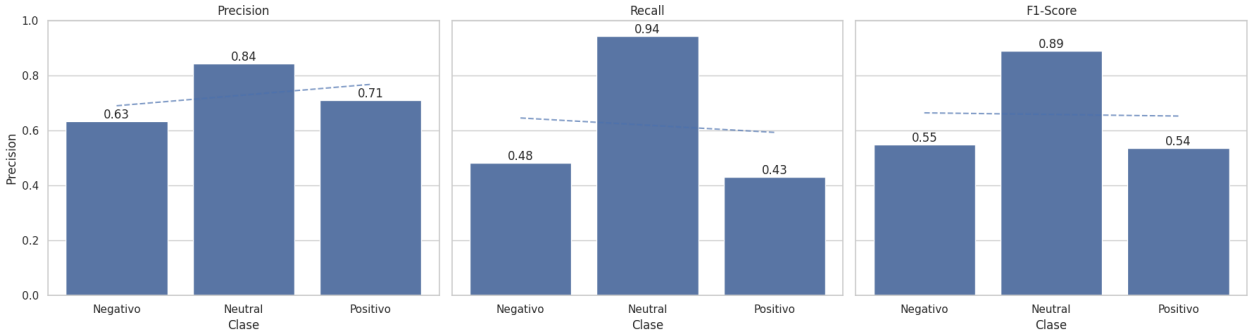


Figura 31. Classification Report

En conjunto, las visualizaciones y métricas confirman que el modelo, tras el ajuste de umbral y calibración, ha logrado un *trade-off* favorable: mantiene una precisión alta en la clase mayoritaria, pero mejora significativamente la detección de "*Negativo*" y "*Positivo*", en línea con la hipótesis inicial. Las decisiones más conservadoras en frases ambiguas,

combinadas con una mejor recuperación de sentimientos explícitos, validan la efectividad del *pipeline* implementado para abordar el desbalance sin sacrificar solidez general.

8.3. Clasificación de resultados obtenidos

Con el propósito de establecer un análisis integral de los modelos evaluados, se desarrolló una clasificación general sobre las métricas de rendimiento obtenidas durante el entrenamiento. Esto se suma al proceso realizado de inferencia anteriormente explicado, el cual permitió identificar no solo qué modelos presentan mejores resultados numéricos, sino también cuáles ofrecen mayor sensibilidad y coherencia interpretativa frente a textos con distintas cargas afectivas.

A continuación, en la Tabla 17, se presenta la matriz comparativa de resultados por métricas. Esta clasificación se realizó en función de los valores alcanzados en las métricas estándar: *Accuracy*, F1-macro y *Recall-macro*, donde el puesto uno corresponde al mejor modelo:

Tabla 17. Comparativo según métricas de desempeño

Posición	Numerador del Experimento	Denominación del Experimento	Accuracy	F1-macro	Recall-macro
1	Quinto	Back-Translation & Focal Oversample	0,83	0,67	0,63
2	Segundo	DistilBETO fine-tuned	0,86	0,60	0,60
3	Tercero	Estrategia focal clases minoritarias con DistilBeto	0,82	0,58	0,58
3	Primero	Entrenamiento múltiple modelos Deep Learning	0,81	0,55	0,56
5	Cuarto	Validación cruzada (5-fold estratificada)	0,86	0,52	0,47

El primer lugar le corresponde al experimento con numerador quinto denominado (*Back-Translation & Focal Oversample*) no solo lidera en métricas cuantitativas, sino también en capacidad inferencial, consolidándose como el enfoque más robusto y equilibrado del estudio. En segundo lugar, se ubicó el modelo del segundo experimento (DistilBETO fine-tuned), que, a pesar de alcanzar el mayor valor de accuracy, mostró una ligera caída en recall y F1, revelando cierta tendencia a favorecer la clase predominante (Neutral) en escenarios ambiguos. El tercer experimento, basado en estrategias focales aplicadas a DistilBETO, mantuvo un buen balance entre las métricas, consolidándose como una opción eficiente para mejorar la sensibilidad sin sacrificar precisión global. Finalmente, el modelo

del cuarto experimento, que implementó una validación cruzada 5-fold sin técnicas de balanceo, obtuvo el rendimiento más limitado, ubicándose en la última posición tanto por métricas como por desempeño cualitativo, especialmente en la detección de emociones menos frecuentes.

9. DISEÑO DE UNA INTERFAZ DE USUARIO PARA EL ANÁLISIS DE SENTIMIENTOS

En este apartado se detalla la sexta fase de CRISP-DM, denominada *Despliegue*, cuyo objetivo principal es desplegar el modelo validado en un entorno de producción para su uso en tareas de análisis de sentimientos. Esta etapa es crucial, ya que permite integrar el modelo en aplicaciones prácticas y garantizar que sus resultados sean accesibles y útiles para los usuarios finales. En este contexto, se desarrolla el quinto objetivo específico de este proyecto aplicado.

El diseño de una interfaz de usuario (*UI*) efectiva para el análisis de sentimientos es fundamental para garantizar que los usuarios finales puedan interactuar de manera intuitiva y eficiente con el sistema [49]. Este capítulo detalla los aspectos clave del diseño e implementación de una interfaz desarrollada con *App Script*, enfocada en facilitar la carga de nuevos registros y la interpretación de los resultados del análisis [50].

La interfaz propuesta busca presentar los datos procesados de manera clara, comprensible y permitir al usuario final incorporar nuevos datos al corpus de análisis de forma estructurada y sin complicaciones técnicas. A continuación, se presentan los detalles de los requerimientos, el flujo de interacción y los componentes de la *UI*, así como las tecnologías empleadas en su desarrollo.

9.1. Requerimientos de diseño

El diseño de la interfaz de usuario para el análisis de sentimientos se centró en satisfacer una serie de requerimientos clave, cuidadosamente definidos para garantizar que el sistema fuera accesible, eficiente y escalable. Estos requisitos no solo respondieron a las necesidades técnicas del proyecto, sino que también priorizaron la experiencia del usuario final, permitiendo una interacción intuitiva y fluida con la herramienta [51].

9.1.1. Intuitividad

La intuitividad se planteó como uno de los objetivos principales del diseño, con el propósito de crear una experiencia de usuario clara, predecible y consistente. El diseño se enfocó en eliminar cualquier complejidad técnica que pudiera ser una barrera para usuarios con conocimientos limitados en análisis de datos.

Para lograr esto, la interfaz adoptó una organización visual limpia y lógica. Las funciones principales, como la carga de archivos y la visualización de resultados, se

dispusieron de manera prominente y accesible, lo que redujo significativamente el tiempo necesario para familiarizarse con el sistema. Además, se incorporaron instrucciones claras y mensajes interactivos para guiar al usuario durante todo el proceso, desde la carga de datos hasta la interpretación de los resultados.

La estructura basada en *App Script* permitió que los elementos visuales se actualizarán dinámicamente, proporcionando retroalimentación inmediata al usuario. Esta característica eliminó la necesidad de recargar la página o realizar acciones adicionales, manteniendo una experiencia fluida y satisfactoria.

9.1.2. Eficiencia

La eficiencia en el procesamiento y visualización de datos fue otro requerimiento central del diseño. La interfaz se desarrolló para aceptar archivos de texto de hasta 200 MB, un límite cuidadosamente establecido para balancear la capacidad de manejo de grandes volúmenes de datos sin comprometer el rendimiento del sistema. Este tamaño de archivo permite analizar transcripciones extensas de conversaciones, asegurando la relevancia del análisis incluso en contextos con datos significativos.

El diseño también consideró la necesidad de procesar y visualizar los resultados en tiempo real. Tan pronto como el usuario carga un archivo y ejecuta el análisis, la interfaz procesa los datos utilizando el modelo preentrenado y muestra los resultados en cuestión de segundos. Este enfoque elimina demoras innecesarias y asegura que los usuarios puedan obtener *insights* prácticos rápidamente.

Además, la visualización de los resultados se estructuró de manera que cada sentimiento clasificado y justificado fuera fácilmente accesible. Esto no solo aceleró la interpretación de los resultados, sino que también permitió que el usuario profundizara en detalles específicos sin perder la visión general de la conversación.

9.1.3. Accesibilidad

Otro aspecto crítico del diseño fue garantizar que la interfaz fuera accesible y comprensible para una amplia variedad de usuarios. Para cumplir con este objetivo, se integraron elementos visuales intuitivos que permitieran identificar rápidamente el sentimiento global de una conversación, utilizando colores y símbolos codificados que destacaran las emociones positivas, neutrales y negativas.

Por ejemplo, los resultados de análisis globales se presentaron en encabezados estilizados con colores específicos: verde para sentimientos positivos, gris para sentimientos neutrales y rojo para sentimientos negativos. Este diseño visual permite a los usuarios obtener una impresión general del análisis con solo un vistazo, sin necesidad de examinar todos los detalles inmediatamente.

9.1.4. Escalabilidad

La escalabilidad fue un factor determinante en el diseño, asegurando que la interfaz pudiera manejar grandes volúmenes de datos a medida que crezcan las necesidades del sistema. Desde la arquitectura del *backend* hasta la estructura visual, cada componente fue diseñado para garantizar que el rendimiento y la experiencia del usuario no se vieran comprometidos, incluso en escenarios de alta demanda.

El uso de *App Script* como *framework* de desarrollo permitió optimizar el manejo de los datos procesados y mantener un flujo estable en la interacción con el usuario. Asimismo, la interfaz se estructuró para que futuras actualizaciones o mejoras pudieran integrarse sin necesidad de rediseñar por completo el sistema.

Por ejemplo, la carga de archivos está preparada para admitir formatos adicionales si se requiere en el futuro, y el procesamiento del modelo se puede ajustar para integrar nuevos algoritmos o parámetros, todo mientras se mantiene la experiencia de usuario actual.

9.2. Flujo de interacción

El flujo de interacción entre el usuario y la interfaz de análisis de sentimientos se diseñó cuidadosamente para garantizar una experiencia intuitiva, fluida y eficiente. Este proceso se estructura en tres etapas principales que cubren desde la carga inicial de los datos hasta la presentación de resultados detallados. Cada etapa del flujo fue desarrollada con el objetivo de maximizar la usabilidad y permitir al usuario obtener *insights* accionables de manera sencilla [51].

9.2.1. Carga del archivo

La interfaz está diseñada para optimizar la experiencia de carga y visualización de archivos de audio y documentos relacionados [ver Figura 32]. Esta funcionalidad clave se distribuye de manera accesible y clara dentro de la aplicación, permitiendo al usuario gestionar su contenido de forma intuitiva y rápida. Los componentes clave incluyen:

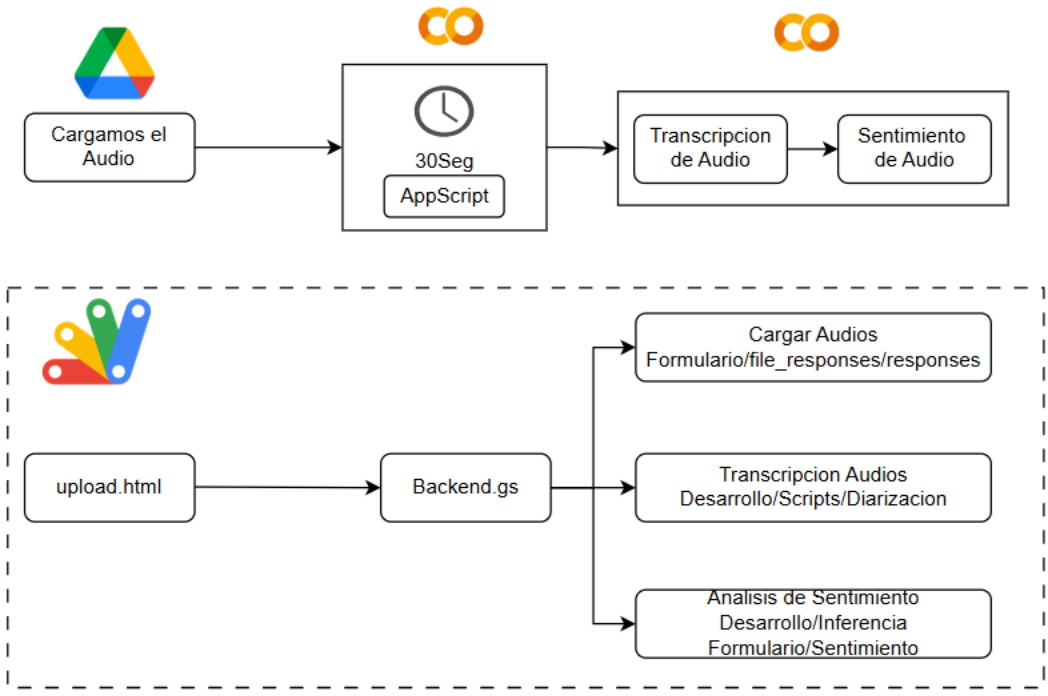


Figura 32. Esquema del diseño implementado en el aplicativo para la carga, transcripción y evaluación de sentimiento de audios de llamadas de Call Center

- **Soporte para archivos .txt:** El sistema admite la carga de archivos en formato de texto, un estándar ampliamente utilizado en transcripciones y datos sin formato. Esta elección facilita la integración de datos provenientes de diferentes fuentes, sin necesidad de preprocesamiento adicional.
- **Gestión de grandes volúmenes de datos:** Se estableció un límite de tamaño por archivo de 200 MB, suficiente para manejar transcripciones extensas de conversaciones, como aquellas que involucran múltiples turnos o análisis detallados. Este límite fue cuidadosamente calibrado para balancear la capacidad del sistema con la velocidad de procesamiento.
- **Integración con Google Drive:** El sistema cuenta con tres botones:
  - *Subir archivo:* El cual se encarga de guardar dentro de una respectiva carpeta de *Google Drive* todos los archivos cargados desde la plataforma.
  - *Audios transcritos:* El cual se encarga de enlistar los archivos *.txt* transcritos dentro de Google Drive, para su descarga inmediata.
  - *Análisis de sentimiento:* Al interactuar con este botón podemos enlistar todos los archivos que contienen las métricas asociadas al audio procesado.

9.2.2. Análisis del contenido

La segunda etapa se activa cuando el usuario hace clic en el botón **"Audios Transcritos"**, ubicado de manera prominente para facilitar su acceso. Este evento desencadena el núcleo del sistema.

- **Procesamiento automatizado**: Una vez que el usuario inicia el análisis, el archivo cargado se procesa automáticamente. El modelo preentrenado se encarga de analizar cada turno de conversación, asignar una categoría de sentimiento (positivo, neutral o negativo) y justificar su clasificación. Este procesamiento se realiza en cuestión de segundos, asegurando que el usuario reciba resultados casi en tiempo real.
- **Interacción sin interrupciones**: Durante el análisis, la interfaz utiliza mensajes interactivos para informar al usuario sobre el progreso. Por ejemplo, una barra de carga o un mensaje como *"Procesando archivo..."* mantiene al usuario informado y reduce la incertidumbre durante esta etapa.

9.2.3. Visualización de resultados

La etapa final del flujo de interacción se enfoca en la presentación de los resultados de manera clara y estructurada. El diseño visual de esta sección fue concebido para resaltar los aspectos clave del análisis y permitir una exploración detallada cuando sea necesario [52].

Visualización del sentimiento global: El sentimiento global de la conversación se presenta en un encabezado estilizado que utiliza colores e íconos distintivos para mejorar la comprensión inmediata:

- **Verde:** Representa un sentimiento positivo, indicando entusiasmo o interés del usuario hacia el tema discutido.
- **Gris:** Denota un sentimiento neutral, asociado a respuestas informativas o sin carga emocional significativa.
- **Rojo:** Indica un sentimiento negativo, reflejando desinterés, frustración o rechazo.

Detalles adicionales por turno: Más allá del sentimiento global, la interfaz presenta los resultados individuales para cada turno de conversación. Estos detalles incluyen:

- El texto transcrito.
- La clasificación de sentimiento asignada.

- Las razones específicas que justifican cada clasificación, como el uso de palabras clave o cambios en el tono emocional.

Cada turno se organiza en secciones expandibles, lo que permite al usuario explorar la conversación a su propio ritmo. Este diseño modular mejora la navegabilidad y evita que la interfaz se sature visualmente en conversaciones largas.

- **Exportación y análisis avanzado:** Para usuarios que requieran un análisis más técnico, la interfaz incluye la opción de exportar los resultados en formato *JSON*. Esto facilita la integración con otras herramientas o la realización de análisis adicionales fuera del sistema.

9.3. Componentes de la interfaz

La interfaz de usuario [ver Figura 33] fue implementada utilizando *App Script*, un *framework* especializado en el desarrollo de aplicaciones interactivas para proyectos de ciencia de datos. Este entorno permitió construir una interfaz dinámica, eficiente y visualmente atractiva, optimizada para procesar, analizar y presentar los resultados del análisis de sentimientos [53]. A continuación, se describen en detalle los componentes principales de la interfaz y su contribución al flujo de interacción del usuario.

Figura 33. Visual de inicio de la interfaz desarrollada con App Script

Lista de archivos: El botón de "*Audios Transcritos"* enlista los archivos de los audios ya transcritos permitiendo tener acceso a la transcripción del audio cargado a un archivo.txt [ver Figura 34].

Figura 34. Lista de archivos transcritos

El diseño estratégico de la barra lateral separa la funcionalidad principal de las áreas de visualización, mejorando la usabilidad y permitiendo que el usuario navegue intuitivamente por las distintas funciones del sistema.

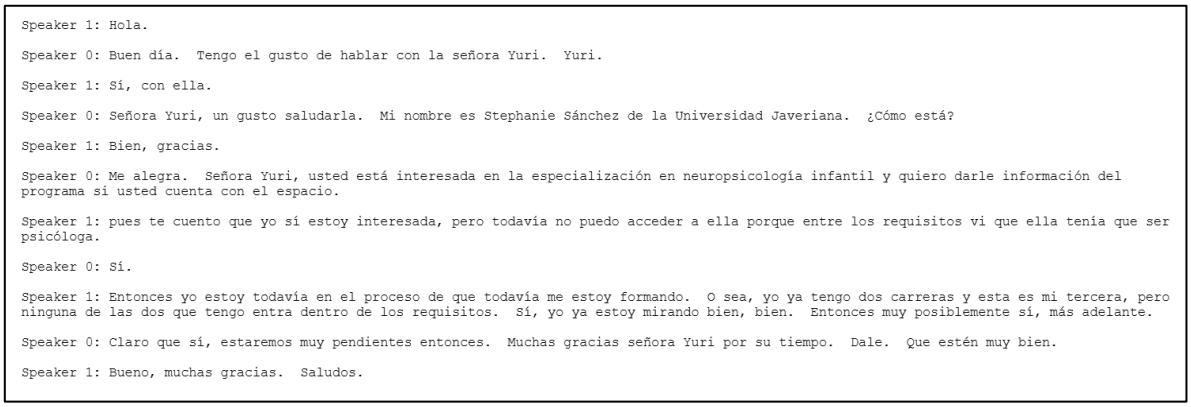
Encabezados estilizados: El diseño visual de la interfaz utiliza encabezados estilizados para destacar información relevante y guiar al usuario durante la interacción. Para lograr esto, se emplearon técnicas de personalización con **CSS** para ajustar colores, tamaños de fuente y formatos.

- Título de la aplicación: El encabezado principal, presentado al inicio de la interfaz, incluye un diseño destacado que define la identidad de la herramienta.
- Resultado del análisis: El sentimiento global de la conversación se muestra en un encabezado grande con colores específicos

El uso de estilos personalizados no solo mejora la estética de la aplicación, sino que también facilita la comprensión de los resultados al resaltar los elementos más importantes.

Visualización de resultados: La presentación de los resultados se diseñó con un enfoque claro y estructurado para que los usuarios puedan interpretar la información sin dificultades. La visualización se organiza en dos niveles principales:

- *Transcripción del audio a Texto:* Como se observó anteriormente el usuario puede tener acceso a los archivos de texto, al acceder a ellos podrá encontrar la transcripción del audio [ver Figura 35].
- *Turnos de conversación:* Cada turno se muestra con su respectivo análisis de sentimiento, permitiendo al usuario explorar los detalles de manera interactiva [ver Figura 36].



Speaker 1: Hola.

Speaker 0: Buen día. Tengo el gusto de hablar con la señora Yuri. Yuri.

Speaker 1: Sí, con ella.

Speaker 0: Señora Yuri, un gusto saludarla. Mi nombre es Stephanie Sánchez de la Universidad Javeriana. ¿Cómo está?

Speaker 1: Bien, gracias.

Speaker 0: Me alegra. Señora Yuri, usted está interesada en la especialización en neuropsicología infantil y quiero darle información del programa si usted cuenta con el espacio.

Speaker 1: pues te cuento que yo sí estoy interesada, pero todavía no puedo acceder a ella porque entre los requisitos vi que ella tenía que ser psicóloga.

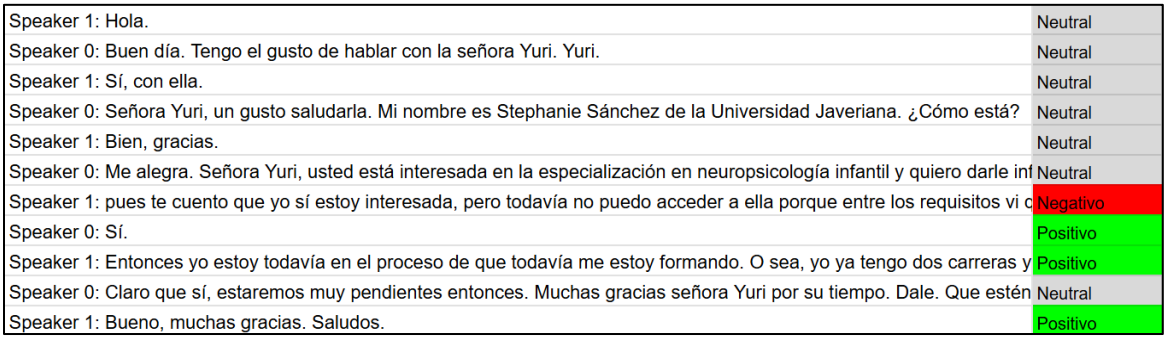
Speaker 0: Sí.

Speaker 1: Entonces yo estoy todavía en el proceso de que todavía me estoy formando. O sea, yo ya tengo dos carreras y esta es mi tercera, pero ninguna de las dos que tengo entra dentro de los requisitos. Sí, yo ya estoy mirando bien, bien. Entonces muy posiblemente sí, más adelante.

Speaker 0: Claro que sí, estaremos muy pendientes entonces. Muchas gracias señora Yuri por su tiempo. Dale. Que estén muy bien.

Speaker 1: Bueno, muchas gracias. Saludos.

Figura 35. Transcripción del audio cargado dentro del aplicativo



Speaker 1: Hola.	Neutral
Speaker 0: Buen día. Tengo el gusto de hablar con la señora Yuri. Yuri.	Neutral
Speaker 1: Sí, con ella.	Neutral
Speaker 0: Señora Yuri, un gusto saludarla. Mi nombre es Stephanie Sánchez de la Universidad Javeriana. ¿Cómo está?	Neutral
Speaker 1: Bien, gracias.	Neutral
Speaker 0: Me alegra. Señora Yuri, usted está interesada en la especialización en neuropsicología infantil y quiero darle inf	Neutral
Speaker 1: pues te cuento que yo sí estoy interesada, pero todavía no puedo acceder a ella porque entre los requisitos vi c	Negativo
Speaker 0: Sí.	Positivo
Speaker 1: Entonces yo estoy todavía en el proceso de que todavía me estoy formando. O sea, yo ya tengo dos carreras y	Positivo
Speaker 0: Claro que sí, estaremos muy pendientes entonces. Muchas gracias señora Yuri por su tiempo. Dale. Que estén	Neutral
Speaker 1: Bueno, muchas gracias. Saludos.	Positivo

Figura 36. Análisis de sentimiento por turno de Speaker

Este enfoque modular mejora la navegabilidad, especialmente en conversaciones largas, al permitir que los usuarios analicen los turnos que consideran más relevantes.

10. CONCLUSIONES Y TRABAJOS FUTUROS

10.1. Conclusiones

En síntesis, este proyecto ha demostrado la complejidad inherente al entrenamiento de modelos de análisis de sentimientos desde cero sobre un corpus pequeño y carente de etiquetas previas. Cada decisión de preprocesamiento —desde la lematización y la eliminación de *stopwords* hasta la *tokenización* y la vectorización *TF–IDF*— tuvo un impacto significativo en la preservación del contexto semántico y en la captura precisa de la señal emocional. Superar este reto implicó no sólo una optimización del uso de recursos computacionales en la transcripción y la diarización de los audios, sino también el diseño de estrategias específicas para la generación de datos sintéticos y el balanceo de clases, con el fin de contrarrestar la escasez y el desbalance de las etiquetas.

Entre las configuraciones experimentadas, el experimento quinto denonimado como ***Back-Translation & Focal Oversample*** se consolidó como la configuración más efectiva y exitosa: combinó traducción inversa para ampliar sintéticamente el corpus, *oversampling* focal de las clases minoritarias, pérdida focal ponderada, ensamblado de un *Bi-LSTM* con un *MLP* sobre vectores *TF–IDF*, y una calibración final mediante *Platt scaling* y ajuste de umbrales. Esta propuesta obtuvo un equilibrio notable entre sensibilidad (*recall*) y precisión global (*accuracy/ROC-AUC*), superando en robustez a la aproximación de *fine-tuning* con *DistilBeto* y estrategia focal. Aunque ningún modelo alcanzó una clasificación perfecta, las métricas conseguidas validan la eficacia de un *pipeline* híbrido que mitiga los sesgos de clase y maximiza el aprendizaje a partir de datos limitados.

Más allá de la contribución académica, esta arquitectura sienta las bases para desplegar en un centro de contacto una herramienta de gestión de calidad automatizada. Al identificar de forma fiable la clasificación de los sentimientos en cada interacción en tiempo real, la dirección de estrategia podrá ajustar guiones, diseñar formaciones específicas y optimizar procesos de atención al cliente. En definitiva, la integración de técnicas clásicas de aprendizaje automático con arquitecturas ligeras de aprendizaje profundo ofrece una solución práctica, escalable y con valor estratégico inmediato para elevar la satisfacción del cliente y mejorar la eficiencia operativa.

10.2. Trabajos Futuros

En este capítulo se identifican las principales actividades futuras derivadas del presente proyecto aplicado. Estas propuestas no responden a falencias en los objetivos alcanzados, sino que reflejan un compromiso con la mejora continua y la ampliación el

impacto del producto mínimo viable (PMV). Consideramos que las funcionalidades implementadas poseen un gran potencial de evolución, lo que permitiría incrementar la propuesta de valor del sistema. A continuación, se detallan las posibles extensiones futuras:

- **Diseño de un módulo de gestión de calidad:** Se plantea el desarrollo de un componente adicional orientado a monitorear el desempeño de los agentes del *call center* y, además, analizar el sentimiento expresado por los clientes durante cada interacción. Esta funcionalidad permitiría mejorar la experiencia del cliente e identificar oportunidades para fortalecer los procesos de formación del personal.
- **Desarrollo de un cuadro de mando integral o *dashboard***: Se propone la creación de un tablero interactivo que integre visualizaciones dinámicas con datos estadísticos y métricas claves extraídas de los archivos procesados. Este recurso facilitaría una gestión basada en datos, aportando claridad en la interpretación de resultados y soporte en la toma de decisiones estratégicas.
- **Actualización del *framework* para entrada multiformato:** Se proyecta adaptar el sistema para que sea capaz de recibir tanto texto como audio como entrada permitiendo un análisis de sentimientos más versátil y aplicable a múltiples casos de uso.
- **Ampliación del conjunto de datos:** Dado que una de las principales limitaciones aquí presentadas fue la cantidad de audios disponibles para el entrenamiento y evaluación de modelos, se propone ampliar progresivamente el corpus para mejorar la generalización y robustez de los modelos implementados.

11. REFERENCIAS

[1] Chaofeng Zhang, Jia Hou, Xueting Tan, Caijuan Chen, Hiroshi Hashimoto, " Enhancing Sentiment Analysis with Collaborative AI: Architecture, Predictions, and Deployment Strategies" *arXiv*, vol. 2410.13247v1, Oct. 2024. Accedido el 16 de diciembre de 2024. [En línea]. Disponible: https://arxiv.org/html/2410.13247v1

[2] "Industria BPO en Colombia entre las más importantes en Latinoamérica". Semana.com Últimas Noticias de Colombia y el Mundo. Accedido el 26 de noviembre de 2023. [En línea]. Disponible: https://www.semana.com/mejor-colombia/articulo/la-industria-de-bpo-en-colombia-es-la-tercera-mas-importante-de-america-latina-de-la-mano-de-la-inversion-extranjera-sigue-creciendo-el-sector/202338/

[3] Parasuraman, A., Zeithaml, V. A., & Berry, L. L. (1988). SERVQUAL: A multiple-item scale for measuring consumer perceptions of service quality. *Journal of Retailing, 64*(1), 12–40

[4] Xue-Yong Fu, C. Chen, Md Tahmid Rahman Laskar, S.Gardiner, P. Hiranandani y, S. Bhushan TN, *Entity-level Sentiment Analysis in Contact Center Telephone Conversations*, arxiv 2022. arXiv: 2210.13401.

[5] Liu, B. (2020). *Sentiment Analysis: Mining Opinions, Sentiments, and Emotions.* Cambridge University Press.

[6] Cambria, E., Das, D., Bandyopadhyay, S., & Feraco, A. (Eds.). (2017). *A practical guide to sentiment analysis* (Vol. 5). Cham: Springer International Publishing.

[7] Jurafsky, D., & Martin, J. H. (2023). *Speech and Language Processing* (3rd ed.). Pearson Education.

[8] Maynard, D., Greenwood, M. A., & Guerini, M. (2014). Who cares about sarcasm? Sentiment annotation on the edge of the frame. *Proceedings of LREC*.

[9] "¿Qué es el aprendizaje automático?" Azure Microsoft. Accedido el 14 de enero de 2024. [En línea]. Disponible: https://azure.microsoft.com/es-es/resources/cloud-computing-dictionary/what-is-machine-learning-platform

[10] LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature, 521*(7553), 436–444.

[11] James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). *An Introduction to Statistical Learning: With Applications in R*. Springer.

[12] Medhat, W., Hassan, A., & Korashy, H. (2014). Sentiment analysis algorithms and applications: A survey. Ain Shams Engineering Journal, 5(4), 1093–1113. https://doi.org/10.1016/j.asej.2014.04.011

[13] Hutto, C., & Gilbert, E. (2014). VADER: A Parsimonious Rule-based Model for Sentiment Analysis of Social Media Text. *Proceedings of the 8th International Conference on Weblogs and Social Media (ICWSM)*, 216–225.

[14] Manning, C. D., Raghavan, P., & Schütze, H. (2008). *Introduction to Information Retrieval*. Cambridge University Press.

[15] Zhang, H. (2004). The optimality of Naive Bayes. *Proceedings of the 17th International Florida Artificial Intelligence Research Society Conference (FLAIRS 2004), 1–6.*

[16] Cortes, C., & Vapnik, V. (1995). Support-vector networks. *Machine Learning, 20*(3), 273–297

[17] Breiman, L. (2001). Random forests. *Machine Learning, 45*(1), 5–32.

[18] Hastie, T., Tibshirani, R., & Friedman, J. (2009). *The Elements of Statistical Learning: Data Mining, Inference, and Prediction* (2nd ed.). Springer.

[19]. 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.

[20] LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature, 521*(7553), 436–444.

[21] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... & Polosukhin, I. (2017). Attention is all you need. *Advances in Neural Information Processing Systems (NeurIPS)*.

[22] Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *NAACL-HLT 2019*.

[23] Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., ... & Amodei, D. (2020). Language models are few-shot learners. *Advances in Neural Information Processing Systems (NeurIPS)*.

[24] Cañete, J., Donoso, S., Bravo-Marquez, F., Carvallo, A., & Araujo, V. (2023). ALBETO and DistilBETO: Lightweight Spanish language models (arXiv:2204.09145). arXiv. https://doi.org/10.48550/arXiv.2204.09145

[25] Graves, A., & Schmidhuber, J. (2005). Framewise phoneme classification with bidirectional LSTM and other neural network architectures. Neural networks, 18(5-6), 602-610.

[26] Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., & Bengio, Y. (2014). Learning phrase representations using RNN encoder-decoder for statistical machine translation. arXiv preprint arXiv:1406.1078.

[27] Wang, Y., Huang, M., Zhu, X., & Zhao, L. (2016, November). *Attention-based LSTM for aspect-level sentiment classification*. In Proceedings of the 2016 conference on empirical methods in natural language processing (pp. 606-615).

[28] M. Desmond, E. Duesterwald, K. Brimijoin, M. Brachman, Qian Pan. *Semi-Automated Data Labeling*, Journal of Machine Learning Research 133:156–169, 2021.

[29] S. M. Quintero. *What Are Word Embeddings? An introduction to the AI that powers language understanding,* Mayo, 2021. Accedido el 14 de enero de 2024. [En línea]. Disponible: https://medium.com/geekculture/what-are-word-embeddings-6f6f677b13ce

[30] A. Boschetti, L. Massaron. Python Data Science Esesentials. 3 Ed. Packt Publishing, 2018. pp. 172 –178.

[31] E. Rønningstad, E. Velldal, L. Øvrelid, *Entity-Level Sentiment Analysis (ELSA): An exploratory task survey*, arxiv 2023. arXiv preprint arXiv: 2304.14241.

[32] S. Gupta. *Sentiment Analysis: Concept, Analysis and Applications,* Enero, 2018. Accedido el 23 de noviembre de 2023. [En línea]. Disponible: https://towardsdatascience.com/sentiment-analysis-concept-analysis-and-applications-6c94d6f58c17

[33] Avi Chawla. *Conversational Sentiment Analysis on Audio Data Analyzing sentiment in Speech,* Julio, 2022. Accedido el 08 de diciembre de 2023. [En línea]. Disponible: https://towardsdatascience.com/conversational-sentiment-analysis-on-audio-data-cd5b9a8e990b#3f04

[34] Shearer, C. (2000). The CRISP-DM Model: The New Blueprint for Data Mining. *Journal of Data Warehousing, 5*(4), 13–22.

[35] Schwaber, K., & Sutherland, J. (2020). *The Scrum Guide.*

[36] Ibrahim, R., & Pradhan, M. (2021). Integrating Agile and Data Mining Methodologies: A Case Study. *International Journal of Computer Science and Information Technologies, 12*(3), 123–130.

[37] Wirth, R., & Hipp, J. (2000). CRISP-DM: Towards a Standard Process Model for Data Mining. *Proceedings of the 4th International Conference on the Practical Applications of Knowledge Discovery and Data Mining*, 29–39.

[38] Kuchaiev, O., et al. (2019). *NeMo: A Toolkit for Building AI Applications using Neural Modules.* NVIDIA.

[39] Python Software Foundation. (2023). *unidecode Documentation.* Retrieved from https://pypi.org/project/Unidecode/

[40] Eisenstein, J. (2019). *Natural Language Processing.* The MIT Press.

[41] Fast, E. (2019). *pyspellchecker Documentation.*

[42] Honnibal, M., & Montani, I. (2017). *spaCy: Industrial-Strength Natural Language Processing in Python.* Explosion AI.

[43] Periñán-Pascual, C., & Arcas-Túnez, F. (2010). The architecture of FunGramKB. *Proceedings of the 7th International Conference on Language Resources and Evaluation (LREC)*, 2667–2674.

[44] Mitrovic, J., Schüller, P., & Samwald, M. (2020). Automated Detection of Bias in Sentiment Analysis Models. *Proceedings of NeurIPS 2020*.

[45] Howard, J., & Gugger, S. (2020). *Deep Learning for Coders with fastai and PyTorch: AI Applications Without a PhD.* O'Reilly Media.

[46] Saari, D. G. (1995). *Basic Geometry of Voting.* Springer Science & Business Media.

[47] Fleiss, J. L. (1971). Measuring Nominal Scale Agreement Among Many Raters. *Psychological Bulletin, 76*(5), 378–382.

[48] Powers, D. M. W. (2011). Evaluation: From Precision, Recall and F-Measure to ROC, Informedness, Markedness & Correlation. *Journal of Machine Learning Technologies, 2*(1), 37–63.

[49] Krug, S. (2014). *Don't Make Me Think, Revisited: A Common Sense Approach to Web Usability.* New Riders.

[50] AppScript Documentation. (2025). Retrieved from https://developers.google.com/apps-script?hl=es-419

[51] Norman, D. A. (2013). *The Design of Everyday Things.* Basic Books.

[52] Few, S. (2006). *Information Dashboard Design: The Effective Visual Communication of Data.* O'Reilly Media.

[53] Shneiderman, B., Plaisant, C., Cohen, M., & Jacobs, S. (2017). *Designing the User Interface: Strategies for Effective Human-Computer Interaction* (6th ed.). Pearson.