UNIVERSIDAD ESTATAL PENÍNSULA DE SANTA ELENA
FACULTAD DE SISTEMAS Y TELECOMUNICACIONES

TITULO DEL TRABAJO DE TITULACIÓN

Detección de neuroticismo en textos en español mediante técnicas de aprendizaje profundo

AUTOR

Reyes Suárez Melany Odalis

PROYECTO DE UNIDAD DE INTEGRACIÓN CURRICULAR

Previo a la obtención del grado académico en
INGENIERO EN TECNOLOGÍAS DE LA INFORMACIÓN

TUTORA

Ing. Lídice Victoria Haz López. Msi.

Santa Elena, Ecuador

Año 2026

UNIVERSIDAD ESTATAL PENÍNSULA DE SANTA ELENA
FACULTAD DE SISTEMAS Y TELECOMUNICACIONES

TRIBUNAL DE SUSTENTACIÓN

Validar únicamente en FirmaEC.
Firmado electrónicamente por:
JOSE MIGUEL SANCHEZ AQUINO

Ing. José Sánchez Aquino. Mgtr.
DIRECTOR DE LA CARRERA

Lidice Victoria Haz Lopez
Time Stamping Security Data

Ing. Lídice Haz López. Mgtr.
TUTORA

Validar únicamente en FirmaEC.
Firmado electrónicamente por:
JAIME BENJAMIN OROZCO IGUASNIA

Ing. Jaime Orozco Iguasnia. Mgtr.
DOCENTE ESPECIALISTA

Validar únicamente en FirmaEC.
Firmado electrónicamente por:
MARJORIE ALEXANDRA CORONEL SUAREZ

Ing. Marjorie Coronel Suárez. Mgtr.
DOCENTE GUÍA UIC

UNIVERSIDAD ESTATAL PENÍNSULA DE SANTA ELENA
FACULTAD DE SISTEMAS Y TELECOMUNICACIONES

CERTIFICACIÓN

Certifico que luego de haber dirigido científica y técnicamente el desarrollo y estructura final del trabajo, este cumple y se ajusta a los estándares académicos, razón por el cual apruebo en todas sus partes el presente trabajo de titulación que fue realizado en su totalidad por REYES SUÁREZ MELANY ODALIS, como requerimiento para la obtención del título de Ingeniero en Tecnologías de la Información.

La Libertad, a los 14 días del mes de Agosto del año 2026

TUTORA

Lidice Victoria Haz Lopez
Time Stamping Security Data

Ing. Lídice Haz López

UNIVERSIDAD ESTATAL PENÍNSULA
DE SANTA ELENA
FACULTAD DE SISTEMAS Y TELECOMUNICACIONES

DECLARACIÓN DE RESPONSABILIDAD

Yo, **Melany Odalis Reyes Suárez**

DECLARO QUE:

El trabajo de Titulación, **Detección de neuroticismo en textos en español mediante técnicas de aprendizaje profundo,** previo a la obtención del título en Ingeniero en Tecnologías de la Información, ha sido desarrollado respetando derechos intelectuales de terceros conforme las citas que constan en el documento, cuyas fuentes se incorporan en las referencias o bibliografías. Consecuentemente este trabajo es de mi total autoría.

En virtud de esta declaración, me responsabilizo del contenido, veracidad y alcance del Trabajo de Titulación referido.

La Libertad, a los 14 días del mes de Agosto del año 2026

EL AUTOR

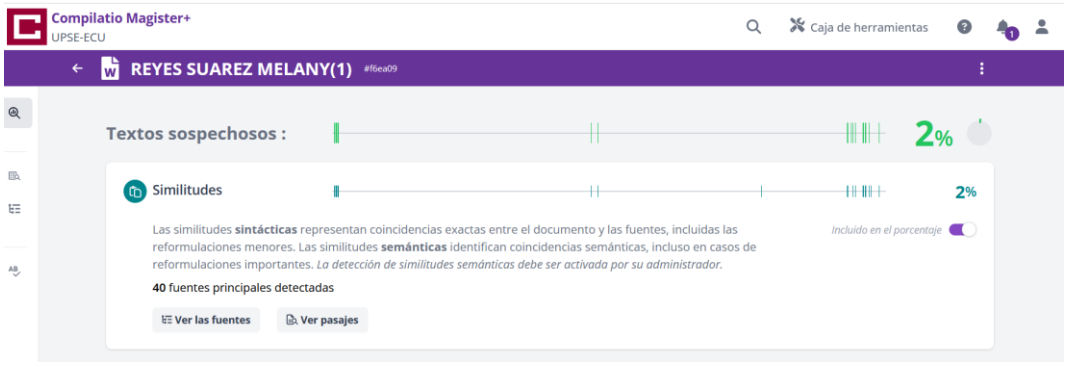
Melany Odalis Reyes Suárez

UNIVERSIDAD ESTATAL PENÍNSULA DE SANTA ELENA

FACULTAD DE SISTEMAS Y TELECOMUNICACIONES

CERTIFICACIÓN DE ANTIPLAGIO

Certifico que después de revisar el documento final del trabajo de titulación denominado, "Detección de neuroticismo en textos en español mediante técnicas de aprendizaje profundo", presentado por el estudiante, Reyes Suárez Melany Odalis fue enviado al Sistema Antiplagio, presentando un porcentaje de similitud correspondiente al 2%, por lo que se aprueba el trabajo para que continúe con el proceso de titulación.



TUTORA

Lidice Victoria Haz Lopez
Time Stamping Security Data

Ing. Lídice Haz López

UNIVERSIDAD ESTATAL PENÍNSULA DE SANTA ELENA
FACULTAD DE SISTEMAS Y TELECOMUNICACIONES

AUTORIZACIÓN

Yo, **Melany Odalis Reyes Suárez**

Autorizo a la Universidad Estatal Península de Santa Elena, para que haga de este trabajo de titulación o parte de él, un documento disponible para su lectura consulta y procesos de investigación, según las normas de la Institución.

Cedo los derechos en línea patrimoniales del presente trabajo de titulación con fines de difusión pública, además apruebo la reproducción dentro de las regulaciones de la Universidad, siempre y cuando esta reproducción no suponga una ganancia económica y se realice respetando mis derechos de autor.

Santa Elena, a los 14 días del mes de Agosto del año 2026

EL AUTOR

Melany Reyes Suárez

AGRADECIMIENTO

Agradezco a Dios por acompañarme durante este camino, darme fortaleza en los momentos difíciles y permitirme culminar una meta que ha requerido años de esfuerzo y perseverancia.

A mis padres, Sr. Daniel Reyes y Sra. Leonor Suárez, por ser el pilar de mi vida, por su amor, sacrificios y apoyo incondicional. A mis hermanos, Kevin, Andy y Justin, a mis sobrinos, Matías y Jair, y a toda mi familia, por creer en mí.

A mi novio, Alonso Reyes, por su amor, paciencia y comprensión; por escucharme, animarme en los momentos difíciles y celebrar conmigo cada avance.

A mi tutora, Ing. Lídice Haz, por su tiempo, orientación y conocimientos; y a mis docentes, por sus enseñanzas y por contribuir a mi formación profesional.

De manera especial, agradezco a la Ing. Mayerli González, gracias por tu amistad, confianza, consejos y por estar siempre dispuesta a ayudarme e impulsarme a seguir adelante.

Finalmente, a Max, Kira y Nina, mis pequeños de cuatro patas, por acompañarme durante tantas horas de estudio.

Melany Odalis, Reyes Suárez

DEDICATORIA

Dedico este logro, con todo mi amor, a mis padres, Sr. Daniel Reyes y Sra. Leonor Suárez, quienes han sido el pilar fundamental de mi vida. Gracias por cada sacrificio, por creer en mí y por enseñarme a seguir adelante aun cuando el camino se vuelve difícil. Todo lo que soy lleva una parte de ustedes y este logro también es suyo.

A mis hermanos, Kevin, Andy y Justin, por su cariño, compañía y por estar presentes en las distintas etapas de mi vida. Crecer junto a ustedes forma parte de quien soy y siempre agradeceré el vínculo que nos une como familia.

A mis sobrinos, Matías y Jair, con todo mi cariño. Deseo que algún día recuerden que los sueños pueden tomar tiempo y que el camino no siempre será sencillo, pero que vale la pena luchar por aquello que realmente desean. Espero verlos alcanzar también cada uno de sus sueños.

Y a Max, Kira y Nina, mis pequeños hijos de cuatro patas y compañeros inseparables, por acompañarme durante tantas horas de estudio y momentos de cansancio. Su cariño hizo más llevadero este camino y también forma parte de este logro.

Melany Odalis, Reyes Suárez

ÍNDICE GENERAL

ÍNDICE DE TABLAS

ÍNDICE DE FIGURAS

RESUMEN

La presente investigación tuvo como objetivo desarrollar un sistema basado en aprendizaje profundo para detectar indicadores lingüísticos asociados al neuroticismo en textos escritos en español. Se trabajó con seis indicadores: vergüenza, irritabilidad, inseguridad, depresión, ira y ansiedad. Para el desarrollo experimental se implementaron los modelos BERT, RoBERTa y DistilBERT mediante *Fine-Tuning*, mientras que FLAN-T5 fue evaluado mediante *Few-Shot Learning*. Los resultados mostraron que BERT obtuvo el mejor desempeño entre los modelos ajustados, alcanzando un F1-score de 0.7622. Por su parte, FLAN-T5 obtuvo un F1-score micro de 0.2602 al ser evaluado sobre 120 comentarios independientes. A partir de los resultados obtenidos, BERT fue seleccionado como modelo principal e integrado en una aplicación para realizar el análisis de comentarios, mientras que FLAN-T5 se incorporó como una alternativa experimental basada en pocos ejemplos.

Palabras claves: neuroticismo, Transformers, aprendizaje profundo.

ABSTRACT

This research aimed to develop a deep learning-based system for detecting linguistic indicators associated with neuroticism in Spanish-written texts. Six indicators were considered: shame, irritability, insecurity, depression, anger, and anxiety. For the experimental development, BERT, RoBERTa, and DistilBERT were implemented through *Fine-Tuning*, while FLAN-T5 was evaluated using *Few-Shot Learning*. The results showed that BERT achieved the best performance among the fine-tuned models, reaching an F1-score of 0.7622. In contrast, FLAN-T5 obtained a micro F1-score of 0.2602 when evaluated on 120 independent comments. Based on the results, BERT was selected as the main model and integrated into an application for comment analysis, while FLAN-T5 was incorporated as an experimental alternative based on a limited number of examples.

Keywords: neuroticism, Transformers, deep learning

INTRODUCCIÓN

El desarrollo de la inteligencia artificial y el Procesamiento del Lenguaje Natural ha permitido ampliar las posibilidades para analizar información expresada mediante textos escritos. En este contexto, los modelos de aprendizaje profundo han adquirido relevancia debido a su capacidad para reconocer patrones lingüísticos y comprender relaciones contextuales presentes en el lenguaje. Estas características han favorecido su aplicación en tareas como la clasificación de textos, el análisis de sentimientos y la identificación de características asociadas a la manera en que las personas expresan emociones a través de medios digitales.

El neuroticismo constituye uno de los rasgos contemplados dentro del modelo de los Cinco Grandes Factores de personalidad y se relaciona con la tendencia a experimentar diferentes estados emocionales negativos. Su identificación a partir del lenguaje representa una tarea compleja, debido a que una misma expresión puede contener diferentes emociones y su significado puede variar de acuerdo con el contexto en el que se utiliza. Esta dificultad adquiere mayor importancia en textos escritos en español, donde las características propias del idioma y la disponibilidad limitada de recursos especializados representan desafíos adicionales para el desarrollo de sistemas automáticos de análisis.

A partir de esta problemática, la presente investigación se orientó al desarrollo de un sistema para la detección automática de indicadores lingüísticos asociados al neuroticismo en textos escritos en español mediante técnicas de aprendizaje profundo. Para ello, se consideraron seis indicadores: vergüenza, irritabilidad, inseguridad, depresión, ira y ansiedad. La tarea fue planteada como un problema de clasificación multietiqueta, permitiendo que un mismo comentario pueda estar relacionado con uno o varios indicadores de manera simultánea.

Durante el desarrollo experimental se implementaron las arquitecturas BERT, RoBERTa y DistilBERT mediante *Fine-Tuning*, con el propósito de analizar su capacidad para adaptarse al conjunto de datos utilizado. Adicionalmente, se incorporó FLAN-T5 mediante *Few-Shot Learning*, empleando instrucciones y un número reducido de ejemplos para orientar el proceso de clasificación. La utilización de ambos enfoques permitió analizar diferentes alternativas para abordar

la identificación automática de los indicadores lingüísticos definidos en la investigación.

La investigación se desarrolló bajo un enfoque cuantitativo, descriptivo y experimental. El proceso comprendió la preparación y preprocesamiento del conjunto de datos, el entrenamiento de los modelos mediante *Fine-Tuning*, la implementación del enfoque *Few-Shot Learning* y la evaluación de los resultados obtenidos. Para medir el desempeño se utilizaron métricas como Accuracy, Precision, Recall, F1-score y Hamming Loss, las cuales permitieron analizar el comportamiento de los modelos dentro de la tarea de clasificación multietiqueta planteada.

Como parte de la propuesta se desarrolló una aplicación que permite ingresar comentarios en español y realizar su análisis mediante el modelo seleccionado a partir de la evaluación experimental. BERT fue integrado como modelo principal para presentar los indicadores detectados y sus respectivas probabilidades, acompañados de una representación gráfica que facilita la interpretación de los resultados. Asimismo, FLAN-T5 fue incorporado como una funcionalidad experimental independiente basada en *Few-Shot Learning*, permitiendo conservar dentro de la propuesta los dos enfoques evaluados durante la investigación.

El documento se encuentra organizado en tres capítulos. El Capítulo 1 presenta la fundamentación de la investigación, incluyendo los antecedentes, descripción del proyecto, objetivos, justificación, alcance, conjunto de datos y metodología empleada. El Capítulo 2 desarrolla los fundamentos teóricos y conceptuales relacionados con el neuroticismo, el Procesamiento del Lenguaje Natural, el aprendizaje profundo, los modelos Transformer y las estrategias utilizadas durante la investigación. Finalmente, el Capítulo 3 describe los componentes de la propuesta, el procesamiento de los datos, la implementación y entrenamiento de los modelos, los resultados obtenidos y las pruebas realizadas sobre la aplicación desarrollada. Posteriormente, se presentan las conclusiones, recomendaciones, referencias bibliográficas y anexos correspondientes.

CAPÍTULO 1. FUNDAMENTACIÓN

1.1. Antecedentes

El crecimiento exponencial del contenido digital generado en plataformas como redes sociales, blogs, foros y otros medios digitales ha propiciado la disponibilidad de grandes volúmenes de información en formato textual. Esta situación ha impulsado el desarrollo de técnicas automáticas para el procesamiento y análisis de dichos datos, consolidando al Procesamiento del Lenguaje Natural (PLN) como una de las áreas de mayor crecimiento dentro de la inteligencia artificial. El PLN permite que los sistemas computacionales comprendan, interpreten y analicen el lenguaje humano, facilitando la identificación de patrones lingüísticos relacionados con emociones, opiniones, comportamientos y otros fenómenos de interés en distintos ámbitos de investigación [1], [2].

El estudio de la personalidad constituye uno de los pilares fundamentales de la psicología, destacándose el modelo de los Cinco Grandes Factores (*Big Five*) como uno de los enfoques con mayor aceptación científica para describir las diferencias individuales. Este modelo considera cinco dimensiones principales de la personalidad: apertura a la experiencia, responsabilidad, extroversión, amabilidad y neuroticismo, siendo esta última especialmente relevante por su asociación con la inestabilidad emocional, la ansiedad, la preocupación constante y la predisposición a experimentar emociones negativas [3]. Diversas investigaciones han validado este modelo en diferentes contextos culturales y poblaciones, consolidándolo como una base teórica ampliamente utilizada en estudios relacionados con el análisis computacional de la personalidad [4].

Desde una perspectiva psicolingüística, numerosas investigaciones han demostrado que el lenguaje refleja características estables de la personalidad. En particular, las personas con altos niveles de neuroticismo tienden a emplear con mayor frecuencia palabras asociadas con emociones negativas, preocupación, inseguridad y estrés, lo que permite inferir determinados rasgos psicológicos a partir del análisis textual [5]. En este contexto, herramientas como *Linguistic Inquiry and Word Count* (*LIWC*) han facilitado la automatización de este proceso mediante la clasificación de

palabras en categorías lingüísticas, emocionales y cognitivas previamente definidas, constituyéndose en uno de los recursos más utilizados en investigaciones sobre análisis del comportamiento humano [6].

En el ámbito computacional, los primeros sistemas de clasificación automática de textos estuvieron basados en técnicas tradicionales de aprendizaje automático, utilizando representaciones del lenguaje como *Bag of Words* (BoW) y la frecuencia de términos (*Term Frequency*). Aunque estos enfoques permitieron importantes avances en tareas de clasificación textual, presentaban limitaciones para representar el significado contextual de las palabras y las relaciones semánticas existentes entre ellas, afectando el desempeño de los modelos en problemas de mayor complejidad [7].

Posteriormente, el desarrollo del aprendizaje profundo permitió superar varias de estas limitaciones mediante arquitecturas neuronales capaces de aprender representaciones distribuidas del lenguaje de manera automática. Entre estos modelos destacan las redes neuronales recurrentes (*Recurrent Neural Networks*, *RNN*) y su variante *Long Short-Term Memory* (LSTM), diseñadas para procesar información secuencial y capturar dependencias temporales en los textos [8]. Sin embargo, a pesar de sus aportes, estos modelos presentan dificultades para modelar dependencias de largo alcance, además de requerir elevados recursos computacionales y tiempos de entrenamiento considerables [9].

Uno de los avance significativo ocurrió con la introducción de la arquitectura Transformer, la cual revolucionó el Procesamiento del Lenguaje Natural mediante el mecanismo de atención (*self-attention*), permitiendo modelar simultáneamente las relaciones existentes entre todas las palabras de un texto sin necesidad de procesarlas de manera secuencial [10]. A partir de esta arquitectura surgieron modelos preentrenados como BERT, capaces de generar representaciones contextuales del lenguaje y alcanzar un desempeño sobresalientes en diversas tareas de clasificación, análisis semántico y comprensión del lenguaje natural [11]. Posteriormente, variantes como RoBERTa y DistilBERT optimizaron el

rendimiento y la eficiencia computacional de estos modelos, ampliando sus posibilidades de aplicación en diferentes dominios del PLN [12], [13].

En este contexto, diversos estudios han aplicado modelos de aprendizaje profundo para la detección automática de rasgos de personalidad a partir del lenguaje. Entre ellos, la investigación desarrollada por Majumder *et al.* propuso un modelo basado en redes neuronales profundas para identificar características de personalidad mediante el análisis de documentos textuales, obteniendo resultados superiores a los alcanzados por métodos tradicionales [14]. Más recientemente, investigaciones basadas en modelos Transformer y grandes modelos de lenguaje (*Large Language Models*, LLM) han demostrado mejoras significativas en la detección automática de rasgos de personalidad, evidenciando la evolución de este campo hacia arquitecturas cada vez más robustas y con mayor capacidad de generalización [15].

En el caso del idioma español, diversos estudios coinciden en que aún existen limitaciones importantes relacionadas con la disponibilidad de conjuntos de datos etiquetados y modelos especializados para esta lengua. Históricamente, la mayor parte de las investigaciones en Procesamiento del Lenguaje Natural se han desarrollado utilizando corpus en idioma inglés, situación que dificulta la transferencia directa de los modelos y puede reducir su desempeño en contextos lingüísticos diferentes [1], [16]. Asimismo, investigaciones recientes sobre clasificación de emociones y análisis de sentimientos en español han evidenciado que la aplicación de modelos basados en Transformers mejora considerablemente la precisión de las predicciones cuando se emplean estrategias adecuadas de preprocesamiento y representación contextual del lenguaje [17], [18].

Otro de los desafíos presentes en las tareas de clasificación automática corresponde a la escasez de datos etiquetados, especialmente en problemas de clasificación multietiqueta como la detección de indicadores del neuroticismo. Frente a esta problemática, las técnicas de aprendizaje Few-Shot han adquirido una creciente relevancia al permitir que los modelos aprendan nuevas tareas utilizando únicamente un número reducido de ejemplos de entrenamiento [19], [20]. En esta línea, el desarrollo de modelos instruccionales como FLAN-T5 ha demostrado que

el ajuste mediante instrucciones (*Instruction Tuning*) mejora considerablemente la capacidad de generalización de los modelos de lenguaje en múltiples tareas de procesamiento textual, incluso cuando la disponibilidad de datos es limitada [21], [22].

En consecuencia, la revisión de los antecedentes evidencia que, aunque se han obtenido avances importantes en la detección automática de rasgos de personalidad mediante técnicas de aprendizaje profundo, todavía persisten desafíos relacionados con la disponibilidad de recursos para el idioma español y con la clasificación multietiqueta de indicadores psicológicos. En este contexto, la presente investigación propone desarrollar un modelo basado en arquitecturas Transformer y técnicas de aprendizaje Few-Shot, utilizando modelos como BERT, RoBERTa, DistilBERT y FLAN-T5, con el propósito de mejorar la detección automática de indicadores del neuroticismo en textos escritos en español y contribuir al desarrollo de herramientas inteligentes para el análisis computacional de la personalidad en textos escritos en español.

1.2. Descripción del Proyecto

En el ámbito del Procesamiento del Lenguaje Natural (PLN), la detección automática de rasgos de personalidad a partir de textos representa un desafío debido a la complejidad del lenguaje humano, caracterizado por su ambigüedad, variabilidad semántica y la coexistencia de múltiples dimensiones emocionales en un mismo contenido. En este contexto, el presente proyecto tuvo como finalidad desarrollar un modelo de aprendizaje profundo para la detección automática de indicadores de neuroticismo en textos escritos en idioma español.

La solución propuesta se fundamentó en la implementación de modelos basados en arquitecturas Transformer, las cuales han demostrado un desempeño sobresaliente en tareas de clasificación de texto gracias a su capacidad para generar representaciones contextuales del lenguaje mediante mecanismos de atención (*self-attention*), facilitando la identificación de patrones lingüísticos complejos asociados a características emocionales y rasgos de personalidad [10], [11].

Para el desarrollo del proyecto se utilizó un conjunto de datos en idioma español obtenido de la plataforma Kaggle. Posteriormente, el conjunto de datos fue sometido a un proceso de preprocesamiento y reetiquetado con el propósito de adecuarlo al problema de investigación y al esquema de clasificación multietiqueta planteado. Como resultado de este proceso, cada texto pudo asociarse simultáneamente a una o varias dimensiones emocionales relacionadas con el neuroticismo, específicamente ansiedad, depresión, inseguridad, vergüenza, ira e irritabilidad.

Debido a estas características, el problema fue abordado como una tarea de clasificación multietiqueta, en la cual una misma instancia puede pertenecer a varias categorías de manera simultánea. Asimismo, durante el análisis del conjunto de datos se identificó un desbalance en la distribución de las clases, condición frecuente en este tipo de problemas y que puede afectar el desempeño de los modelos de aprendizaje profundo [23]. Para reducir este efecto, durante el entrenamiento se aplicó una estrategia de ponderación de la función de pérdida (*Weighted Loss*), permitiendo mejorar el aprendizaje de las clases menos representadas [24], [25].

El desarrollo del modelo contempló diversas etapas. Inicialmente, se realizó el preprocesamiento de los textos mediante técnicas de limpieza, normalización, tokenización, eliminación de palabras vacías y lematización, con el objetivo de obtener información más consistente para el entrenamiento del modelo [26], [27]. Posteriormente, se generaron representaciones contextuales del lenguaje utilizando modelos Transformer preentrenados, permitiendo capturar las relaciones semánticas presentes en cada texto [10], [11].

En la fase de entrenamiento se implementaron y compararon los modelos BERT, RoBERTa y DistilBERT, adaptándolos a la tarea de clasificación multietiqueta de indicadores de neuroticismo [14]. El proceso de entrenamiento y evaluación se desarrolló en el entorno Google Colab, aprovechando el uso de unidades de procesamiento gráfico (GPU) para optimizar los tiempos de ejecución y facilitar la experimentación [19], [7], [37].

Adicionalmente, se incorporó un enfoque de aprendizaje Few-Shot mediante el modelo FLAN-T5, con el propósito de evaluar la capacidad de generalización del sistema en escenarios con disponibilidad limitada de datos etiquetados. Este enfoque permitió analizar el comportamiento del modelo utilizando un número reducido de ejemplos, condición que resulta de gran interés en aplicaciones reales donde la obtención de grandes volúmenes de datos anotados puede representar un desafío [28], [18], [29].

La evaluación del modelo se realizó mediante métricas ampliamente utilizadas en problemas de clasificación multietiqueta, entre ellas Accuracy, Precisión, Recall, F1-Score y Hamming Loss, permitiendo analizar el rendimiento del sistema desde diferentes perspectivas y comparar objetivamente el desempeño de los modelos implementados [30], [25].

Como complemento al desarrollo del modelo, se implementó una interfaz de usuario que permite ingresar textos escritos en español y visualizar automáticamente los indicadores de neuroticismo detectados. Esta interfaz facilita la interacción con el sistema y evidencia su funcionamiento en un entorno práctico, contribuyendo a demostrar la aplicabilidad de la solución desarrollada.

Finalmente, el proyecto se desarrolló dentro de la línea de investigación Tecnología y Sistemas de la Información (TSI), específicamente en la sublínea Inteligencia Computacional, aportando al desarrollo de herramientas basadas en inteligencia artificial para la detección automática de indicadores de neuroticismo en textos escritos en español. Asimismo, la investigación permitió integrar técnicas modernas de procesamiento del lenguaje natural y aprendizaje profundo para abordar un problema de clasificación multietiqueta, contribuyendo al fortalecimiento de aplicaciones orientadas al análisis automatizado de información textual y al desarrollo de soluciones tecnológicas en el ámbito académico e investigativo.

1.3. Objetivos del Proyecto

1.3.1. Objetivo General

Desarrollar un modelo basado en técnicas de aprendizaje profundo para la detección automática de indicadores lingüísticos de neuroticismo en textos en español, mediante el uso de arquitecturas Transformer y estrategias N-Shot, con el fin de mejorar la precisión y capacidad de generalización en tareas de clasificación psicológica multietiqueta.

1.3.2. Objetivo Específicos

- Construir un dataset en español mediante datasets disponibles en Kaggle y reetiquetarlo con indicadores de neuroticismo.
- Aplicar técnicas de procesamiento textual, asegurando la limpieza, normalización, tokenización y lematización de los datos.
- Diseñar modelos de aprendizaje profundo basados en arquitecturas Transformer para la clasificación multietiqueta de rasgos de neuroticismo.
- Incorporar técnicas de aprendizaje N-Shot (Few-Shot Learning) con el propósito de mejorar la capacidad de generalización del modelo en escenarios con datos limitados.
- Evaluar el desempeño del modelo propuesto mediante métricas de clasificación multietiqueta, tales como precisión, recall, F1-score y Hamming Loss, analizando los resultados para identificar patrones lingüísticos asociados al neuroticismo, con el fin de validar su efectividad.

1.4. Justificación del Proyecto

La presente investigación surge ante la necesidad de desarrollar herramientas automatizadas capaces de analizar rasgos de personalidad a partir del lenguaje natural, especialmente en entornos digitales donde la producción de texto es constante. En estos espacios, las personas expresan ideas, emociones y estados de ánimo que pueden ser estudiados mediante técnicas computacionales, permitiendo obtener información relevante sobre patrones lingüísticos asociados al comportamiento y al estado emocional de las personas. En este sentido, el análisis

del lenguaje no se limita únicamente al contenido explícito, sino que también permite identificar patrones emocionales presentes en los textos [23].

El desarrollo de este proyecto resulta pertinente al proponer el uso de modelos basados en arquitecturas Transformer junto con técnicas de aprendizaje N-Shot (Few-Shot Learning), lo cual permite abordar problemas de clasificación multietiqueta en escenarios donde la disponibilidad de datos es limitada. Este enfoque representa una alternativa frente a métodos tradicionales, ya que mejora la capacidad de los modelos para comprender el contexto del lenguaje y adaptarse a nuevas situaciones [19]. Además, al centrarse en textos en idioma español, se contribuye a disminuir la brecha existente en la investigación, considerando que gran parte de los avances en Procesamiento del Lenguaje Natural se han desarrollado en idioma inglés.

La detección de indicadores de neuroticismo en textos puede tener diversas aplicaciones en contextos reales. Por ejemplo, en el ámbito educativo podría contribuir como herramienta de apoyo para el análisis del estado emocional de los estudiantes, mientras que en plataformas digitales permitiría estudiar grandes volúmenes de información generada por los usuarios. Asimismo, este tipo de herramientas puede servir como apoyo en procesos relacionados con el bienestar emocional, aportando información complementaria para la toma de decisiones, sin sustituir las evaluaciones realizadas por profesionales.

Desde una perspectiva técnica, el uso de modelos Transformer permite capturar relaciones complejas dentro del lenguaje, facilitando la identificación de patrones lingüísticos asociados a estados emocionales. A su vez, la incorporación de técnicas N-Shot permite evaluar el comportamiento del modelo en escenarios con disponibilidad limitada de datos etiquetados, condición frecuente en aplicaciones reales [29]. De esta manera, se promueve el uso de enfoques más flexibles dentro del campo de la inteligencia artificial aplicada al análisis del lenguaje.

Asimismo, la presente investigación se encuentra alineada con el Plan Nacional de Desarrollo Ecuador No Se Detiene 2025–2029, el cual constituye un instrumento

estratégico para la planificación y ejecución de políticas públicas orientadas al desarrollo sostenible, la innovación y la transformación digital del país [31]. Este plan contempla diversos ejes enfocados en mejorar la calidad de vida de la población, fortalecer el desarrollo económico y promover el uso de tecnologías emergentes como apoyo a la gestión del conocimiento y la toma de decisiones.

En este contexto, el proyecto se relaciona con el eje Desarrollo Económico, Productivo y Empleo, debido a que impulsa el uso de herramientas basadas en inteligencia artificial y Procesamiento del Lenguaje Natural para el análisis automatizado de información textual, fomentando la innovación tecnológica y el desarrollo de soluciones digitales aplicadas a problemas reales [31]. La implementación de modelos de aprendizaje profundo y arquitecturas Transformer contribuye al fortalecimiento de competencias tecnológicas y al avance de la investigación en áreas estratégicas del sector tecnológico.

De igual manera, esta investigación se vincula con el Eje Social, ya que plantea una herramienta orientada al análisis de indicadores lingüísticos asociados al neuroticismo, la cual puede servir como apoyo en ámbitos como la salud emocional, la educación y la psicología, permitiendo identificar patrones relacionados con el bienestar emocional de las personas [31]. De esta manera, el proyecto no solo aporta al desarrollo tecnológico, sino también al fortalecimiento del bienestar social mediante el uso responsable de la inteligencia artificial.

1.5. Alcance del Proyecto

El presente proyecto comprendió el desarrollo de un modelo basado en técnicas de aprendizaje profundo para la detección automática de indicadores de neuroticismo en textos escritos en idioma español, considerando un conjunto específico de dimensiones emocionales asociadas a este rasgo de personalidad. Para ello, se implementaron modelos basados en arquitecturas Transformer y estrategias de aprendizaje N-Shot, orientados a la clasificación multietiqueta de textos. El desarrollo del sistema permitió analizar patrones lingüísticos relacionados con el neuroticismo, contribuyendo a la aplicación de técnicas de inteligencia artificial en

el procesamiento del lenguaje natural y al fortalecimiento de investigaciones en este ámbito.

El alcance del sistema abarcó el procesamiento de datos textuales provenientes de un dataset en español, el cual fue sometido a etapas de preprocesamiento, representación del lenguaje y entrenamiento de modelos de clasificación multietiqueta. Se implementaron y evaluaron modelos basados en arquitecturas Transformer, específicamente BERT, RoBERTa y DistilBERT, para identificar las dimensiones de ansiedad, depresión, inseguridad, vergüenza, ira e irritabilidad presentes en los textos.

Asimismo, el proyecto incluyó la implementación de un enfoque de aprendizaje N-Shot, mediante el modelo FLAN-T5, con el propósito de evaluar la capacidad de generalización del sistema en escenarios con disponibilidad limitada de datos etiquetados. La evaluación del desempeño de los modelos se realizó mediante métricas de clasificación multietiqueta, permitiendo analizar objetivamente la efectividad de cada arquitectura en la detección de los indicadores definidos.

Como parte del alcance del proyecto, se desarrolló una interfaz de usuario que permite la interacción con el modelo implementado, facilitando la realización de pruebas mediante el ingreso de textos en idioma español y la visualización de los resultados obtenidos. El desarrollo e implementación del sistema se llevó a cabo en el entorno Google Colab, utilizando herramientas y librerías especializadas en Procesamiento del Lenguaje Natural y aprendizaje profundo.

No obstante, el proyecto presenta ciertas limitaciones. En primer lugar, el análisis se restringe a textos en idioma español, por lo que no considera otros idiomas. En segundo lugar, el modelo se basa en un conjunto específico de dimensiones emocionales asociadas al neuroticismo, por lo que no abarca la totalidad de los rasgos de personalidad. Adicionalmente, el desempeño del modelo depende de la calidad y cantidad de los datos disponibles, así como del desbalance presente en el conjunto de datos utilizado para el entrenamiento. Finalmente, el sistema desarrollado no busca reemplazar las evaluaciones

psicológicas realizadas por profesionales, ya que el análisis de la personalidad requiere criterios clínicos y contextuales especializados. Su propósito consiste en servir como una herramienta de apoyo para la identificación automatizada de patrones lingüísticos asociados al neuroticismo en textos escritos en español. Mediante técnicas de Procesamiento del Lenguaje Natural y aprendizaje profundo, el sistema permite analizar grandes volúmenes de información de forma rápida y eficiente, proporcionando resultados complementarios para fines académicos e investigativos.

1.5.1 Conjunto de datos

El conjunto de datos utilizado en la presente investigación estuvo conformado por textos escritos en idioma español obtenidos a partir de un dataset disponible en la plataforma Kaggle. Los registros fueron seleccionados debido a que contienen información textual susceptible de ser analizada mediante técnicas de Procesamiento del Lenguaje Natural para la identificación de indicadores lingüísticos asociados al neuroticismo. La utilización de este recurso permitió disponer de una base de información adecuada para el desarrollo y evaluación de los modelos de aprendizaje profundo implementados durante la investigación.

Como parte de la preparación del conjunto de datos, los comentarios fueron sometidos a un proceso de revisión y reetiquetado con el propósito de adaptarlos al problema de clasificación multietiqueta planteado. A partir de este procedimiento se definieron seis indicadores lingüísticos: vergüenza, irritabilidad, inseguridad, depresión, ira y ansiedad. Cada comentario pudo asociarse con una o varias de estas categorías de manera simultánea, debido a que un mismo texto puede contener diferentes expresiones emocionales relacionadas con el rasgo de neuroticismo.

El conjunto de datos presentó una distribución desigual entre las diferentes etiquetas, evidenciando la existencia de desbalance entre las categorías consideradas. Esta característica fue tomada en cuenta durante el entrenamiento de los modelos BERT, RoBERTa y DistilBERT mediante la aplicación de una función de pérdida ponderada, con el propósito de reducir el efecto de la diferencia existente

entre la cantidad de registros correspondientes a cada indicador y favorecer un proceso de aprendizaje más equilibrado.

Para el entrenamiento y evaluación de los modelos ajustados mediante Fine-Tuning, el conjunto de datos fue dividido en dos subconjuntos independientes, destinando el 80 % para entrenamiento y el 20 % para prueba. Esta división permitió utilizar una parte de los comentarios para el ajuste de los parámetros de los modelos y reservar un conjunto independiente para evaluar posteriormente su capacidad de generalización mediante las métricas definidas en la investigación.

En el caso particular de FLAN-T5, debido a que fue implementado mediante la estrategia Few-Shot Learning, se estableció una organización diferente de los datos. Se utilizaron 13 comentarios como ejemplos dentro del prompt, 50 comentarios como conjunto de Desarrollo y 120 comentarios independientes como conjunto de Pruebas. Esta separación permitió definir el procedimiento de decisión utilizando únicamente el conjunto de Desarrollo y reservar los 120 comentarios finales exclusivamente para evaluar el comportamiento del modelo sobre información no utilizada durante la configuración del enfoque Few-Shot.

1.6. Metodología del Proyecto

1.6.1. Metodología de Investigación

La metodología de investigación aplicada en el presente proyecto adoptó un enfoque cuantitativo, descriptivo y experimental, orientado al análisis de datos textuales mediante técnicas de aprendizaje profundo. La combinación de estos enfoques permitió abordar el problema desde diferentes perspectivas, integrando el análisis estadístico, la descripción de las características del conjunto de datos y la evaluación práctica de los modelos desarrollados. Esta metodología proporcionó una base sólida para el desarrollo, entrenamiento y validación de los modelos utilizados en la detección automática de indicadores de neuroticismo en textos escritos en español.

Enfoque cuantitativo

El enfoque cuantitativo se centró en la medición de variables numéricas relacionadas con el desempeño de los modelos en la tarea de clasificación

multietiqueta. La evaluación de las técnicas implementadas se realizó mediante métricas como precisión, *recall*, F1-score y *Hamming Loss*, las cuales permitieron comparar objetivamente los resultados obtenidos e identificar el modelo con mejor desempeño para el problema planteado [12]. Asimismo, durante el proceso experimental se efectuó un análisis estadístico del rendimiento de cada modelo, proporcionando evidencia cuantitativa sobre su capacidad para detectar indicadores de neuroticismo.

Enfoque descriptivo

El enfoque descriptivo se empleó en las etapas iniciales de la investigación con el propósito de analizar las características de las técnicas de procesamiento del lenguaje natural y aprendizaje profundo utilizadas en la clasificación de textos. En este contexto, se describieron las particularidades de las arquitecturas implementadas, así como su funcionamiento en la identificación de patrones lingüísticos asociados al neuroticismo [12]. De igual manera, se analizaron las características del conjunto de datos, considerando la distribución de las clases, la naturaleza multietiqueta del problema y el desbalance existente entre las categorías.

Enfoque experimental

El enfoque experimental orientó la implementación y evaluación de los modelos de aprendizaje profundo desarrollados durante la investigación. Para ello, se realizaron pruebas controladas con el objetivo de medir el desempeño de las arquitecturas Transformer y de la estrategia de aprendizaje Few-Shot implementada mediante FLAN-T5. Los resultados obtenidos permitieron comparar el comportamiento de cada modelo y realizar los ajustes necesarios durante el proceso de entrenamiento, favoreciendo la selección de la alternativa con mejor rendimiento para la detección automática de indicadores de neuroticismo [12].

1.6.2. Beneficiarios del Proyecto

Los beneficiarios del presente proyecto comprendieron tanto usuarios directos como indirectos, quienes pueden aprovechar los resultados obtenidos mediante el desarrollo del modelo de detección automática de indicadores de neuroticismo en textos escritos en español. La implementación de esta herramienta constituye un

aporte para diferentes áreas relacionadas con el procesamiento del lenguaje natural, el aprendizaje profundo y el análisis computacional de información textual, favoreciendo el desarrollo de futuras investigaciones y aplicaciones basadas en inteligencia artificial.

Los beneficiarios directos correspondieron a estudiantes, investigadores y profesionales vinculados a las áreas de inteligencia artificial, procesamiento del lenguaje natural y ciencia de datos, quienes pueden utilizar el modelo desarrollado como una herramienta de apoyo para el análisis automatizado de textos en español. Asimismo, la investigación proporciona una base metodológica que puede servir como referencia para el desarrollo de nuevos modelos orientados a la clasificación multietiqueta y al estudio de rasgos de personalidad mediante técnicas de aprendizaje profundo.

Por otra parte, los beneficiarios indirectos incluyeron áreas como la psicología, la educación y el análisis del comportamiento humano, donde el sistema propuesto puede contribuir como herramienta complementaria para el análisis automatizado de patrones emocionales presentes en textos digitales. De igual manera, los resultados obtenidos pueden servir de apoyo para futuras investigaciones interdisciplinarias que integren técnicas de inteligencia artificial con el estudio del comportamiento y las emociones expresadas mediante el lenguaje escrito.

1.6.3. Variables

En la presente investigación se consideraron las variables necesarias para evaluar la relación entre la aplicación de modelos de aprendizaje profundo y su desempeño en la detección automática de indicadores de neuroticismo en textos escritos en español. La definición de estas variables permitió orientar el desarrollo del estudio y establecer los elementos fundamentales para la evaluación de los resultados obtenidos durante el proceso experimental.

Variable independiente: Modelos de aprendizaje profundo basados en arquitecturas Transformer y técnicas de aprendizaje N-Shot.

Variable dependiente: Desempeño del modelo en la clasificación multietiqueta de indicadores de neuroticismo en textos escritos en español.

1.6.4. Análisis de recolección de datos

1.6.4.1 Técnicas de Recolección de Información

La recolección de datos se llevó a cabo mediante la utilización de un conjunto de datos en idioma español obtenido de la plataforma Kaggle, el cual constituyó la base para el desarrollo, entrenamiento y evaluación del modelo propuesto. Este dataset fue seleccionado por contener información textual relacionada con indicadores de neuroticismo y por su pertinencia para la implementación de modelos de aprendizaje profundo orientados a la clasificación multietiqueta. Asimismo, el procesamiento de la información se realizó en el entorno Google Colab, el cual proporcionó los recursos computacionales necesarios para ejecutar los modelos y desarrollar los experimentos efectuados durante la investigación.

El conjunto de datos estuvo conformado por 8112 comentarios escritos en idioma español, los cuales fueron sometidos al proceso de preparación y reetiquetado definido en la investigación. La cantidad total de registros empleados se presenta en la Tabla 1. Posteriormente, los comentarios fueron clasificados de acuerdo con las dimensiones emocionales establecidas para el estudio, cuya distribución se presenta en la Tabla 2. Esta organización permitió disponer de un conjunto de datos adecuado para el entrenamiento y la evaluación de los modelos implementados.

Dataset	
Comentarios	8112

Tabla 1. Cantidad de comentarios del dataset. Elaboración propia

Distribución de los datos	
Emociones	**Cantidad**
DEPRESIÓN	4740
IRRITABILIDAD	4035
VERGÜENZA	3968

Distribución de los datos	
Emociones	**Cantidad**
INSEGURIDAD	3123
IRA	2886
ANSIEDAD	1746

Tabla 2. Distribución de comentarios por dimensión emocional. Elaboración propia

La Tabla 2 presenta la distribución de los comentarios de acuerdo con las dimensiones emocionales consideradas en la investigación. Se observa un desbalance entre las categorías del conjunto de datos, siendo depresión la dimensión con mayor número de registros, mientras que ansiedad corresponde a la categoría con menor cantidad de comentarios. Esta característica fue considerada durante el entrenamiento de los modelos mediante la aplicación de pesos en la función de pérdida, con el propósito de favorecer un aprendizaje más equilibrado y mejorar el desempeño en la clasificación de las categorías menos representadas.

1.6.4.2 Técnicas de Recolección de Información

El conjunto de datos fue sometido a un proceso de preparación con el propósito de transformarlo en un formato adecuado para su análisis mediante modelos de aprendizaje profundo. Estas actividades permitieron mejorar la calidad de la información, reducir la presencia de elementos irrelevantes y optimizar la representación de los textos para el entrenamiento de las arquitecturas Transformer implementadas en la investigación. Asimismo, este procedimiento contribuyó a obtener un conjunto de datos consistente con los objetivos planteados y con las características propias de un problema de clasificación multietiqueta.

Inicialmente, se realizó la limpieza de los datos, eliminando caracteres especiales, signos de puntuación, emojis y otros elementos que no aportaban información relevante para el análisis. Posteriormente, se llevó a cabo la tokenización, mediante la cual los textos fueron segmentados en unidades léxicas que facilitaron su procesamiento por parte de los modelos de lenguaje. Estas etapas permitieron

normalizar la información y preparar adecuadamente los textos para las fases posteriores del procesamiento.

A continuación, se efectuó la representación del lenguaje utilizando modelos basados en arquitecturas Transformer, los cuales generaron representaciones contextuales capaces de capturar relaciones semánticas complejas entre las palabras y mejorar la comprensión del contenido textual [10]. Posteriormente, para afrontar el desbalance existente entre las clases, se aplicó una estrategia de ajuste en la función de pérdida mediante el uso de pesos (*Weighted Loss*), favoreciendo un aprendizaje más equilibrado durante el entrenamiento de los modelos.

Finalmente, se implementó una estrategia de aprendizaje N-Shot mediante el modelo FLAN-T5, con el propósito de evaluar la capacidad de generalización del sistema en escenarios con una cantidad limitada de ejemplos de entrenamiento. Esta técnica permitió analizar el comportamiento del modelo frente a datos con escasa disponibilidad de muestras etiquetadas y complementar la evaluación realizada con las arquitecturas Transformer empleadas en la investigación.

1.7. Metodología de desarrollo

El desarrollo del modelo computacional se fundamentó en metodologías de aprendizaje automático y aprendizaje profundo, las cuales comprendieron las etapas de preparación de datos, representación del lenguaje, entrenamiento, validación y evaluación de los modelos, siguiendo los principios establecidos en la literatura especializada [33]. En este sentido, el proyecto se estructuró en fases que permitieron organizar de manera sistemática el proceso de implementación, garantizando la calidad del conjunto de datos, la correcta configuración de los modelos y la evaluación de su desempeño durante las diferentes etapas de la investigación.

La metodología de desarrollo permitió establecer un flujo de trabajo ordenado para la ejecución del proyecto, iniciando con la preparación y el procesamiento del conjunto de datos, seguido por el entrenamiento de los modelos BERT, RoBERTa y DistilBERT mediante la estrategia de *Fine-Tuning*, así como la implementación del modelo FLAN-T5 utilizando aprendizaje Few-Shot. Finalmente, los modelos

fueron evaluados mediante diferentes métricas de desempeño, permitiendo comparar sus resultados y determinar la alternativa con mejor capacidad para la detección automática de indicadores de neuroticismo en textos escritos en español.

La Figura 1 presenta de manera general las fases que conformaron la metodología de desarrollo implementada durante la investigación, mostrando la secuencia de actividades realizadas desde la preparación de los datos hasta la evaluación de los modelos de aprendizaje profundo. Esta estructura facilitó la preparación del conjunto de datos, la implementación de los modelos de aprendizaje profundo y la evaluación de su desempeño, asegurando la coherencia del proceso metodológico y el cumplimiento de los objetivos planteados para la investigación

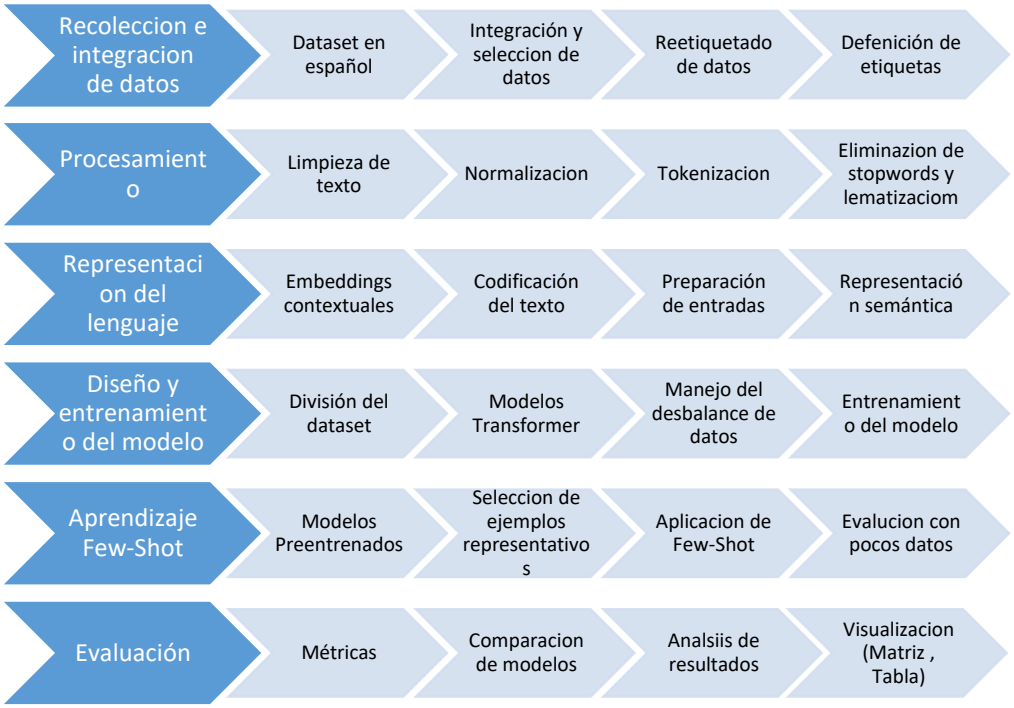


Figura 1. Metodología de desarrollo. Elaboración propia.

CAPÍTULO 2. PROPUESTA

2.1. Marco Contextual

El acelerado desarrollo de la inteligencia artificial ha transformado significativamente la forma en que se procesan y analizan los datos en entornos digitales, especialmente aquellos generados en formato textual. Este crecimiento ha sido impulsado por la masificación de plataformas digitales, como redes sociales, foros y sistemas de comunicación en línea, donde los usuarios producen grandes volúmenes de información que reflejan opiniones, emociones y comportamientos. En este contexto, el procesamiento automatizado del lenguaje humano se ha convertido en un componente esencial para la extracción de conocimiento a partir de datos no estructurados, favoreciendo el desarrollo de aplicaciones en ámbitos como la educación, la psicología y el análisis del comportamiento humano [32].

El Procesamiento del Lenguaje Natural (PLN) constituye una de las áreas más relevantes de la inteligencia artificial, orientada a la interacción entre las personas y los sistemas computacionales mediante el lenguaje. Su evolución ha permitido el desarrollo de modelos capaces de comprender, interpretar y generar texto con elevados niveles de precisión, facilitando tareas como la clasificación semántica, la detección de emociones y el análisis de rasgos de personalidad. Estos avances han sido impulsados por las técnicas de aprendizaje profundo, las cuales permiten modelar relaciones complejas dentro del lenguaje y superar las limitaciones de los enfoques tradicionales [33].

En este escenario, el análisis computacional de la personalidad ha adquirido una creciente importancia como un campo interdisciplinario que integra conocimientos de la psicología y la inteligencia artificial. Este enfoque parte de la premisa de que el lenguaje constituye una manifestación del estado emocional y cognitivo de las personas, permitiendo inferir rasgos de personalidad a partir de patrones lingüísticos presentes en los textos. En particular, el neuroticismo ha sido ampliamente estudiado debido a su estrecha relación con la inestabilidad emocional, lo que lo convierte en un rasgo apropiado para su análisis mediante técnicas de Procesamiento del Lenguaje Natural en entornos digitales [34].

El presente proyecto se desarrolló en el contexto académico de la Universidad Estatal Península de Santa Elena (UPSE), institución que promueve la investigación aplicada y el uso de tecnologías emergentes para el desarrollo de soluciones innovadoras. La implementación del modelo se realizó en un entorno controlado de experimentación utilizando Google Colab, plataforma que proporcionó acceso a recursos computacionales en la nube, incluyendo unidades de procesamiento gráfico (GPU). Este entorno facilitó la implementación, el entrenamiento y la evaluación de los modelos de aprendizaje profundo, además de favorecer la reproducibilidad de los resultados obtenidos.

Asimismo, el proyecto respondió a la necesidad de desarrollar soluciones adaptadas al idioma español, considerando que una gran parte de los modelos de procesamiento del lenguaje natural han sido entrenados principalmente con información en inglés. Esta situación representa un desafío para las aplicaciones desarrolladas en español, debido a las diferencias lingüísticas, culturales y contextuales que pueden influir en el desempeño de los modelos. En este sentido, la investigación contribuye al desarrollo de herramientas capaces de analizar textos en español de manera eficiente, especialmente en tareas de clasificación multietiqueta orientadas a la detección de indicadores de neuroticismo.

2.2. Marco Conceptual

2.2.1. Fundamentos de la personalidad

La personalidad constituye uno de los principales campos de estudio de la psicología, debido a que permite comprender las diferencias individuales en la forma de pensar, sentir y actuar de las personas. Larsen y Buss explican que la personalidad está conformada por un conjunto de características psicológicas relativamente estables que distinguen a cada individuo y orientan su manera de percibir el entorno, relacionarse con los demás y responder ante distintas situaciones. Su estudio ha dado lugar al desarrollo de diversas teorías y modelos que buscan explicar el comportamiento humano desde una perspectiva científica y sistemática [35].

Entre los diferentes enfoques propuestos para el estudio de la personalidad, las teorías basadas en rasgos han adquirido una amplia aceptación debido a su capacidad para describir las diferencias individuales mediante dimensiones observables y relativamente estables. En este contexto, Costa y McCrae contribuyeron al fortalecimiento del Modelo de los Cinco Grandes Factores (Big Five), considerado actualmente uno de los marcos teóricos con mayor respaldo empírico para el análisis de la personalidad. Dentro de este modelo, el neuroticismo representa una de las dimensiones más estudiadas por su estrecha relación con la regulación emocional y las respuestas frente al estrés [3].

La comprensión de la personalidad requiere analizar las características que distinguen a cada individuo y que influyen en su comportamiento. Estas características son estudiadas mediante el enfoque de los rasgos de personalidad, el cual proporciona una base para explicar las diferencias individuales y ha permitido el desarrollo de modelos ampliamente utilizados en la investigación psicológica, como el Modelo de los Cinco Grandes Factores (Big Five) [4].

2.2.1.1 Personalidad

La personalidad es un constructo psicológico que engloba el conjunto de características relativamente estables que distinguen a un individuo y determinan su forma de pensar, sentir y comportarse. Estas características permiten explicar las diferencias individuales en la manera en que las personas perciben su entorno, interactúan con los demás y afrontan diversas situaciones a lo largo de la vida. Larsen y Buss señalan que la personalidad constituye una organización dinámica de procesos psicológicos que influyen de manera constante en la conducta humana y en la adaptación al medio [35].

El desarrollo de la personalidad resulta de la interacción entre factores biológicos, psicológicos, sociales y ambientales, los cuales contribuyen a la formación de patrones relativamente estables de comportamiento. Costa y McCrae sostienen que, aunque estos patrones pueden experimentar cambios derivados de las experiencias personales y del contexto, conservan una estabilidad suficiente para diferenciar a los individuos a lo largo del tiempo. Esta estabilidad ha permitido que la

personalidad sea estudiada mediante modelos científicos orientados a describir y analizar sus principales dimensiones [3].

El estudio de la personalidad ha favorecido la construcción de diversos enfoques teóricos destinados a comprender las diferencias individuales desde una perspectiva científica. Entre ellos, las teorías basadas en rasgos ocupan un lugar destacado, debido a que describen la personalidad mediante características relativamente estables que pueden observarse, evaluarse y compararse entre individuos. Este enfoque constituye uno de los fundamentos más utilizados en la investigación psicológica contemporánea y sirve de base para el desarrollo de modelos ampliamente aceptados, como el Modelo de los Cinco Grandes Factores [4].

2.2.1.2 Rasgos de la personalidad

Los rasgos de la personalidad constituyen características relativamente estables que describen la manera habitual en que un individuo piensa, siente y actúa frente a distintas situaciones. A diferencia de los estados emocionales, que pueden variar según el contexto o las experiencias inmediatas, los rasgos representan tendencias duraderas que permiten explicar las diferencias individuales en el comportamiento. De acuerdo con Larsen y Buss, los rasgos facilitan la descripción y comparación de las personas mediante patrones consistentes que se mantienen a lo largo del tiempo [35].

El estudio de los rasgos ha permitido desarrollar modelos que organizan la personalidad en dimensiones específicas, facilitando su evaluación e interpretación dentro del ámbito científico. Costa y McCrae sostienen que los rasgos poseen un grado considerable de estabilidad y constituyen la base para comprender cómo las personas responden ante diferentes estímulos, interactúan con su entorno y desarrollan determinados patrones conductuales. Esta perspectiva ha favorecido la construcción de instrumentos psicométricos ampliamente utilizados para evaluar la personalidad en diversos contextos [3].

La clasificación de los rasgos de personalidad ha contribuido al desarrollo de modelos teóricos con un sólido respaldo empírico, entre los cuales destaca el Modelo de los Cinco Grandes Factores (Big Five). Este modelo organiza los rasgos

en cinco dimensiones fundamentales que describen de manera integral la personalidad humana y sirven como referencia para numerosas investigaciones en psicología de la personalidad [4].

2.2.1.3 Modelo Big Five

El Modelo de los Cinco Grandes Factores, conocido internacionalmente como Big Five, constituye uno de los enfoques más reconocidos y respaldados para el estudio científico de la personalidad. Su desarrollo se fundamenta en investigaciones basadas en el análisis factorial de los rasgos de personalidad, las cuales permitieron identificar cinco dimensiones amplias capaces de describir las principales diferencias individuales observadas entre las personas. Gracias a su solidez teórica y empírica, este modelo ha sido ampliamente adoptado en investigaciones psicológicas realizadas en diferentes culturas y contextos [4].

Las cinco dimensiones propuestas por este modelo son apertura a la experiencia (*Openness*), responsabilidad (*Conscientiousness*), extraversión (*Extraversion*), amabilidad (*Agreeableness*) y neuroticismo (*Neuroticism*). Cada una representa un conjunto de características psicológicas que permiten describir tendencias relativamente estables del comportamiento humano. En conjunto, estas dimensiones ofrecen una visión integral de la personalidad, facilitando su estudio mediante instrumentos psicométricos diseñados para evaluar cada rasgo de manera independiente [3].

Diversas investigaciones han demostrado que el Modelo Big Five presenta altos niveles de validez y confiabilidad para describir la personalidad, razón por la cual se ha convertido en un referente dentro de la psicología contemporánea. Su aplicación se ha extendido a diferentes áreas del conocimiento, como la psicología clínica, la psicología organizacional, la educación y, en los últimos años, al análisis computacional del comportamiento humano mediante técnicas de inteligencia artificial y procesamiento del lenguaje natural. Esta versatilidad ha favorecido su utilización en estudios orientados a identificar patrones psicológicos a partir de información textual [35], [4].

Entre las cinco dimensiones que conforman el modelo, el neuroticismo ha despertado un especial interés debido a su relación con la estabilidad emocional y la tendencia a experimentar emociones negativas. Las características asociadas a este rasgo permiten comprender diferencias individuales en la respuesta ante situaciones de estrés, incertidumbre y presión emocional, motivo por el cual constituye una de las dimensiones más investigadas dentro del Modelo Big Five [3].

2.2.1.4 Neuroticismo

El neuroticismo es una de las cinco dimensiones fundamentales del Modelo Big Five y describe la tendencia de una persona a experimentar emociones negativas con mayor frecuencia e intensidad que otros individuos. Este rasgo se asocia con la susceptibilidad al estrés, la preocupación constante, la ansiedad, la inseguridad y la inestabilidad emocional. Según Lahey, el neuroticismo representa una predisposición general hacia respuestas emocionales intensas ante situaciones percibidas como amenazantes o adversas, constituyendo uno de los rasgos más relevantes para comprender las diferencias individuales en el funcionamiento emocional [36].

Las personas con niveles elevados de neuroticismo suelen presentar mayor sensibilidad frente a eventos cotidianos, interpretando determinadas situaciones como más estresantes o difíciles de afrontar. Esta característica puede manifestarse mediante emociones como preocupación, tristeza, irritabilidad, vergüenza o frustración. Widiger señala que, aunque el neuroticismo no constituye un trastorno psicológico, niveles elevados de este rasgo pueden incrementar la vulnerabilidad al desarrollo de diferentes alteraciones emocionales cuando interactúan con factores ambientales y personales [37].

Desde la psicología de la personalidad, el neuroticismo ha sido ampliamente investigado debido a su capacidad para explicar diferencias en la regulación emocional y en la forma en que las personas afrontan experiencias positivas y negativas. Diversos estudios han evidenciado que este rasgo mantiene una relación consistente con variables como el bienestar psicológico, el estrés percibido y la

adaptación social, razón por la cual continúa siendo una de las dimensiones con mayor interés dentro de la investigación científica sobre la personalidad [38].

Finalmente, el estudio del neuroticismo ha trascendido el ámbito tradicional de la psicología y ha despertado interés en disciplinas como la lingüística computacional y la inteligencia artificial. En estos campos, el análisis de patrones lingüísticos permite identificar indicadores asociados a este rasgo de personalidad, favoreciendo el desarrollo de investigaciones orientadas a comprender la relación entre el lenguaje y los procesos psicológicos sin sustituir la evaluación realizada mediante instrumentos especializados [34].

2.2.2. Fundamentos del Procesamiento del lenguaje Natural

El Procesamiento del Lenguaje Natural (PLN) constituye una de las principales áreas de la inteligencia artificial orientadas al estudio y tratamiento computacional del lenguaje humano. Su desarrollo integra conocimientos provenientes de disciplinas como la lingüística, la informática, la psicología cognitiva y el aprendizaje automático, con el propósito de diseñar sistemas capaces de comprender, interpretar y generar información expresada en lenguaje natural. Gracias a estos avances, el PLN se ha convertido en una herramienta fundamental para el análisis automatizado de grandes volúmenes de información textual [1] [39].

El crecimiento de los contenidos digitales y la disponibilidad de datos generados por los usuarios han impulsado el desarrollo de técnicas cada vez más sofisticadas para analizar textos de forma automática. En la actualidad, el PLN tiene aplicaciones en áreas como la traducción automática, los asistentes virtuales, los sistemas de recomendación, el análisis de sentimientos, la recuperación de información y la clasificación de documentos. Estas aplicaciones demuestran la importancia de esta disciplina en la solución de problemas relacionados con el procesamiento de información lingüística [2].

La evolución del Procesamiento del Lenguaje Natural también ha favorecido la integración de modelos de aprendizaje profundo y arquitecturas basadas en Transformer, los cuales han incrementado significativamente la capacidad de los sistemas para comprender el contexto y el significado del lenguaje. Para

comprender estos avances resulta necesario revisar previamente algunos fundamentos relacionados con la psicolingüística, la ciberpsicología y las principales técnicas utilizadas durante el procesamiento de textos.

2.2.2.1 Psicolingüística

La psicolingüística es una disciplina interdisciplinaria que estudia la relación entre el lenguaje y los procesos cognitivos involucrados en su producción, comprensión y adquisición. Su objetivo principal consiste en explicar cómo las personas procesan la información lingüística y de qué manera los mecanismos mentales intervienen durante la comunicación. Este campo integra conocimientos provenientes de la psicología y la lingüística para comprender el funcionamiento del lenguaje desde una perspectiva científica [40].

El estudio de la psicolingüística ha permitido comprender que el lenguaje no solo constituye un medio para transmitir información, sino también una manifestación de procesos cognitivos y emocionales que reflejan la forma en que los individuos interpretan su realidad. La selección de palabras, la estructura de las oraciones y las expresiones utilizadas durante la comunicación pueden proporcionar información acerca de aspectos relacionados con el pensamiento, la memoria, la atención y las emociones [41].

Actualmente, la psicolingüística proporciona fundamentos teóricos para diversas investigaciones relacionadas con el análisis automatizado del lenguaje, favoreciendo el desarrollo de herramientas capaces de estudiar patrones lingüísticos presentes en textos escritos. Su aporte resulta especialmente relevante en áreas como la inteligencia artificial y el Procesamiento del Lenguaje Natural, donde la comprensión del lenguaje humano constituye un elemento esencial para el desarrollo de modelos computacionales [1].

2.2.2.2 Ciberpsicología

La ciberpsicología es una rama de la psicología que estudia la interacción entre las personas y las tecnologías digitales, analizando cómo el uso de internet, las redes

sociales y otros entornos virtuales influye en los procesos cognitivos, emocionales y conductuales. Esta disciplina busca comprender la manera en que los individuos construyen relaciones, expresan emociones y desarrollan comportamientos dentro de los espacios digitales, considerando que las tecnologías han transformado significativamente las formas de comunicación e interacción social [42].

El desarrollo de la comunicación digital ha generado grandes volúmenes de información textual que reflejan opiniones, experiencias, emociones y patrones de comportamiento de los usuarios. En este contexto, la ciberpsicología ha permitido estudiar cómo las características psicológicas pueden manifestarse mediante el lenguaje utilizado en plataformas digitales, proporcionando un marco teórico para comprender la expresión de rasgos de personalidad y estados emocionales en los entornos virtuales. Estas investigaciones han demostrado que las interacciones en línea constituyen una valiosa fuente de información para el análisis del comportamiento humano [43].

En la actualidad, la ciberpsicología mantiene una estrecha relación con disciplinas como el Procesamiento del Lenguaje Natural y la inteligencia artificial, debido a que el análisis automatizado de textos facilita el estudio de patrones lingüísticos presentes en la comunicación digital. La integración de estas áreas ha favorecido el desarrollo de investigaciones orientadas a comprender el comportamiento humano mediante técnicas computacionales, consolidando el uso del lenguaje escrito como una fuente relevante para el análisis de características psicológicas [44].

2.2.2.3 Procesamiento del Lenguaje Natural (PLN)

El Procesamiento del Lenguaje Natural (PLN) es una disciplina de la inteligencia artificial que desarrolla métodos y técnicas para que los sistemas computacionales puedan comprender, interpretar y generar lenguaje humano de manera automática. Su propósito consiste en reducir la brecha entre la comunicación humana y los sistemas informáticos, permitiendo que las computadoras analicen información expresada en lenguaje natural de forma eficiente. Para alcanzar este objetivo, el PLN integra conocimientos provenientes de la lingüística, la informática, la

inteligencia artificial y el aprendizaje automático, lo que ha favorecido el desarrollo de aplicaciones capaces de procesar grandes volúmenes de información textual [1].

El crecimiento de la información disponible en formato digital ha impulsado el desarrollo de técnicas de PLN orientadas al análisis automatizado de textos. Entre sus principales aplicaciones se encuentran la clasificación de documentos, el análisis de sentimientos, la traducción automática, el reconocimiento de entidades, los sistemas de preguntas y respuestas, los asistentes virtuales y el resumen automático de información. Estas aplicaciones han permitido que el PLN se convierta en una de las áreas con mayor desarrollo dentro de la inteligencia artificial, debido a su capacidad para transformar información no estructurada en conocimiento útil para diferentes ámbitos científicos y tecnológicos [2].

La evolución del Procesamiento del Lenguaje Natural ha estado estrechamente relacionada con los avances en el aprendizaje profundo y el desarrollo de nuevas arquitecturas para el procesamiento de secuencias textuales. En los últimos años, los modelos basados en Transformer han mejorado significativamente la capacidad de los sistemas para comprender el contexto y las relaciones semánticas presentes en los textos, marcando un cambio significativo en la forma en que las computadoras representan, comprenden y procesan el lenguaje natural. Estos avances impulsaron el desarrollo de modelos preentrenados capaces de adaptarse a múltiples tareas mediante procesos de ajuste fino y técnicas de aprendizaje con un número reducido de ejemplos [45].

2.2.2.4 Aprendizaje profundo

El aprendizaje profundo (*Deep Learning*) es una rama del aprendizaje automático que utiliza redes neuronales artificiales con múltiples capas para aprender representaciones complejas a partir de grandes volúmenes de datos. A diferencia de los métodos tradicionales, estas redes son capaces de extraer automáticamente características relevantes de la información de entrada, reduciendo la necesidad de diseñar manualmente atributos específicos para cada problema. Gracias a esta capacidad, el aprendizaje profundo ha impulsado avances significativos en áreas

como la visión por computadora, el reconocimiento de voz y el procesamiento del lenguaje natural [46].

El funcionamiento del aprendizaje profundo se basa en arquitecturas compuestas por capas interconectadas de neuronas artificiales, las cuales transforman progresivamente la información mediante operaciones matemáticas y funciones de activación. Durante el proceso de entrenamiento, el modelo ajusta sus parámetros internos utilizando algoritmos de optimización que minimizan el error entre las predicciones y los resultados esperados. Este mecanismo permite que las redes neuronales aprendan patrones complejos y relaciones no lineales presentes en los datos, mejorando progresivamente su capacidad de generalización [47].

En el ámbito del Procesamiento del Lenguaje Natural, el aprendizaje profundo ha permitido desarrollar modelos capaces de comprender relaciones sintácticas y semánticas con un nivel de precisión superior al alcanzado por los enfoques tradicionales. La incorporación de arquitecturas neuronales avanzadas ha favorecido la representación contextual del lenguaje y ha impulsado el desarrollo de modelos preentrenados que constituyen la base de muchas aplicaciones actuales. Entre estos avances destacan las arquitecturas Transformer, las cuales marcaron un cambio significativo en el procesamiento automático del lenguaje y dieron origen a modelos ampliamente utilizados en tareas de comprensión y generación de texto [45].

2.2.2.5 Tokenización

La tokenización es una de las primeras etapas del preprocesamiento en el Procesamiento del Lenguaje Natural y consiste en dividir un texto en unidades más pequeñas denominadas *tokens*. Estas unidades pueden corresponder a palabras, caracteres, subpalabras o signos de puntuación, dependiendo del método de tokenización y de los requerimientos de la tarea que se desea realizar. Este proceso facilita que los sistemas computacionales transformen el lenguaje natural en una representación estructurada que pueda ser interpretada y procesada por los modelos de aprendizaje automático [26].

La selección del método de tokenización influye directamente en el desempeño de los modelos de Procesamiento del Lenguaje Natural, ya que determina la forma en que la información lingüística será representada durante el entrenamiento y la inferencia. En la actualidad, los modelos basados en Transformer emplean técnicas de tokenización por subpalabras, las cuales permiten representar palabras desconocidas o poco frecuentes mediante la combinación de unidades lingüísticas más pequeñas. Esta estrategia contribuye a reducir el tamaño del vocabulario y mejora la capacidad de los modelos para comprender diferentes variantes del lenguaje[48].

La tokenización constituye un paso esencial dentro del flujo de procesamiento textual, debido a que prepara la información para las etapas posteriores de normalización, análisis lingüístico y generación de representaciones semánticas. Su adecuada implementación favorece la conservación del contexto y mejora la calidad de las representaciones utilizadas por los modelos modernos de Procesamiento del Lenguaje Natural [1].

2.2.2.6 Lematización

La lematización es un proceso lingüístico utilizado en el Procesamiento del Lenguaje Natural que consiste en transformar las palabras flexionadas en su forma base o lema, considerando su categoría gramatical y el contexto en el que aparecen. A diferencia de otros métodos de normalización textual, la lematización preserva el significado de las palabras al aplicar reglas lingüísticas y recursos léxicos que permiten obtener una representación más precisa del lenguaje. Este procedimiento facilita el análisis computacional de los textos al reducir la variabilidad causada por las diferentes formas que puede adoptar una misma palabra [49].

La aplicación de la lematización contribuye a disminuir la complejidad del vocabulario presente en un corpus, agrupando bajo un mismo lema palabras que comparten el mismo significado léxico. Por ejemplo, términos como *estudiar*, *estudia*, *estudiando* y *estudiaron* pueden asociarse al lema estudiar, favoreciendo una representación más uniforme de la información. Esta normalización mejora la

calidad de los datos utilizados durante el análisis textual y facilita la identificación de patrones lingüísticos en diferentes tareas de Procesamiento del Lenguaje Natural [1].

La lematización constituye una etapa importante dentro del preprocesamiento de textos, especialmente en aplicaciones relacionadas con clasificación documental, minería de texto y análisis semántico. Al conservar el significado lingüístico de las palabras, este procedimiento favorece una representación más consistente de la información y contribuye a mejorar el desempeño de diversos modelos utilizados en el análisis automatizado del lenguaje [1].

2.2.2.7 Embeddings

Los embeddings son representaciones vectoriales de las palabras que permiten transformar información textual en valores numéricos que pueden ser interpretados por los modelos de aprendizaje automático. A diferencia de las representaciones tradicionales, donde cada palabra se considera una unidad independiente, los embeddings ubican las palabras dentro de un espacio vectorial continuo, en el cual aquellas que poseen significados o contextos similares se representan mediante vectores cercanos entre sí. Esta característica permite capturar relaciones semánticas y sintácticas presentes en el lenguaje natural, favoreciendo una representación más rica de la información textual [50].

El desarrollo de los embeddings representó un avance importante para el Procesamiento del Lenguaje Natural, ya que permitió superar las limitaciones de métodos tradicionales basados en frecuencias de palabras, como Bag of Words o TF-IDF, los cuales no consideran el contexto ni las relaciones semánticas entre los términos. Posteriormente surgieron modelos como Word2Vec, GloVe y FastText, capaces de aprender representaciones distribuidas de las palabras a partir de grandes colecciones de textos. Estos enfoques mejoraron significativamente el desempeño de diversas tareas relacionadas con clasificación de textos, análisis semántico y recuperación de información [51], [52].

La evolución de los embeddings condujo al desarrollo de representaciones contextuales, en las cuales el significado de una palabra depende del contexto en el que aparece dentro de una oración. Este avance permitió el surgimiento de modelos basados en arquitecturas Transformer, como BERT y sus variantes, que generan representaciones dinámicas capaces de capturar relaciones lingüísticas más complejas. Actualmente, los embeddings contextuales constituyen uno de los principales fundamentos del Procesamiento del Lenguaje Natural moderno y han contribuido significativamente al desarrollo de sistemas con mayor capacidad de comprensión del lenguaje humano [53].

2.2.3. Arquitecturas y modelos basados en Transformer

Las arquitecturas basadas en Transformer representan uno de los avances más importantes en la evolución del Procesamiento del Lenguaje Natural. Su aparición permitió superar diversas limitaciones presentes en arquitecturas neuronales anteriores, como las Redes Neuronales Recurrentes (RNN) y las Redes de Memoria a Largo Corto Plazo (LSTM), las cuales procesaban la información de manera secuencial y presentaban dificultades para modelar dependencias de largo alcance en los textos. La incorporación del mecanismo de atención permitió que los modelos analizaran simultáneamente todas las palabras de una secuencia, favoreciendo una comprensión más eficiente del contexto lingüístico [10].

El desarrollo de estas arquitecturas impulsó la creación de modelos preentrenados capaces de aprender representaciones generales del lenguaje a partir de grandes colecciones de textos. Posteriormente, dichos modelos pueden adaptarse a tareas específicas mediante procesos de ajuste fino (*fine-tuning*) o estrategias de aprendizaje con pocos ejemplos (*few-shot learning*), reduciendo significativamente la necesidad de entrenar modelos desde cero. Esta capacidad ha favorecido su adopción en tareas como clasificación de textos, traducción automática, resumen de documentos, respuesta a preguntas y generación de lenguaje natural [54].

La evolución de las arquitecturas Transformer ha dado origen a numerosos modelos especializados que presentan diferentes estrategias de entrenamiento y optimización. Entre los más representativos se encuentran BERT, RoBERTa,

DistilBERT y FLAN-T5, los cuales han demostrado un desempeño sobresaliente en diversas tareas de Procesamiento del Lenguaje Natural y constituyen algunos de los modelos más utilizados en la investigación actual [55].

2.2.3.1 Arquitectura Transformer

La arquitectura Transformer es un modelo de red neuronal propuesto por Vaswani et al. en 2017 con el objetivo de mejorar el procesamiento de secuencias y superar las limitaciones de las arquitecturas recurrentes empleadas hasta ese momento. A diferencia de las Redes Neuronales Recurrentes (RNN) y las Redes de Memoria a Largo Corto Plazo (LSTM), el Transformer procesa todos los elementos de una secuencia de forma paralela, lo que reduce significativamente el tiempo de entrenamiento y mejora la capacidad del modelo para capturar dependencias entre palabras que pueden encontrarse muy distantes dentro de un texto. Esta característica representó un cambio importante en el desarrollo de modelos para el Procesamiento del Lenguaje Natural [10].

El funcionamiento del Transformer se fundamenta en el mecanismo de autoatención (*self-attention*), el cual permite que cada palabra de una secuencia determine la importancia de las demás palabras presentes en el mismo contexto. Gracias a este mecanismo, el modelo puede asignar diferentes niveles de atención a cada término durante el procesamiento del texto, favoreciendo una mejor comprensión de las relaciones sintácticas y semánticas existentes entre las palabras. Adicionalmente, la arquitectura incorpora un sistema de codificación posicional (*positional encoding*), cuya función consiste en conservar la información relacionada con el orden de las palabras dentro de la secuencia, aspecto fundamental debido a que el procesamiento paralelo no considera de forma natural la posición de cada elemento [10].

La arquitectura Transformer está compuesta por un codificador (*encoder*) y un decodificador (*decoder*), ambos formados por múltiples capas de atención y redes neuronales totalmente conectadas. El codificador transforma el texto de entrada en representaciones contextualizadas, mientras que el decodificador utiliza dicha información para generar la salida correspondiente en tareas como traducción

automática o generación de texto. Aunque numerosos modelos actuales utilizan únicamente una de estas dos estructuras, la arquitectura original estableció las bases para el desarrollo de modelos preentrenados que han revolucionado el Procesamiento del Lenguaje Natural, entre ellos BERT, RoBERTa, DistilBERT y FLAN-T5 [56].

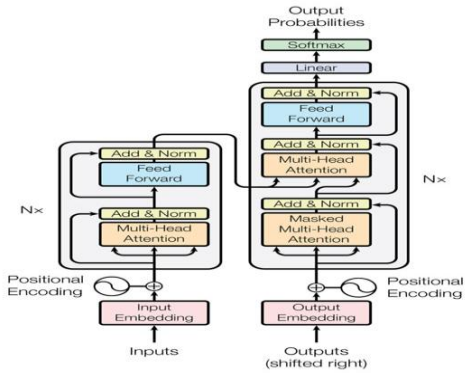


Figura 2. Arquitectura general del modelo Transformer.[10]

2.2.3.2 Modelos basados en Transformer

Los modelos basados en Transformer son arquitecturas de aprendizaje profundo desarrolladas a partir del modelo propuesto por Vaswani et al. (2017), cuya finalidad es comprender y generar lenguaje natural mediante mecanismos de atención. Estos modelos aprovechan el aprendizaje obtenido durante una etapa de preentrenamiento sobre grandes volúmenes de datos textuales, lo que les permite adquirir conocimiento general del lenguaje antes de ser adaptados a tareas específicas. Gracias a este enfoque, han logrado mejorar significativamente el rendimiento en diversas aplicaciones del Procesamiento del Lenguaje Natural, reduciendo además el tiempo y los recursos necesarios para entrenar modelos desde cero [55].

Una de las principales características de estos modelos es la posibilidad de reutilizar el conocimiento aprendido mediante estrategias como el ajuste fino (*fine-tuning*), en el cual un modelo previamente entrenado se adapta a una tarea concreta utilizando un conjunto de datos más reducido. Asimismo, los avances recientes han permitido incorporar técnicas como el aprendizaje con pocos ejemplos (*few-shot*

learning), donde el modelo es capaz de resolver nuevas tareas a partir de un número limitado de ejemplos o instrucciones. Estas capacidades han favorecido su aplicación en problemas de clasificación de textos, análisis de sentimientos, reconocimiento de entidades, traducción automática, resumen de documentos y generación de lenguaje natural [57].

El continuo desarrollo de los modelos basados en Transformer ha dado lugar a diferentes arquitecturas, cada una diseñada para optimizar aspectos específicos relacionados con el entrenamiento, el rendimiento computacional o la capacidad de generalización. Entre las más representativas se encuentran BERT, orientado principalmente a la comprensión bidireccional del lenguaje; RoBERTa, que introduce mejoras en el proceso de entrenamiento de BERT; DistilBERT, diseñado para reducir el costo computacional manteniendo un desempeño competitivo; y FLAN-T5, basado en el aprendizaje mediante instrucciones para mejorar la capacidad de generalización en múltiples tareas. Estas arquitecturas constituyen referentes en la investigación actual y representan la base de numerosas aplicaciones modernas de inteligencia artificial aplicada al lenguaje [54].

2.2.3.3 BERT

BERT (*Bidirectional Encoder Representations from Transformers*) es un modelo de lenguaje preentrenado desarrollado por Devlin et al. en 2019 con el propósito de mejorar la comprensión contextual del lenguaje natural mediante una arquitectura basada exclusivamente en el codificador (*encoder*) del Transformer. Su principal innovación consiste en el aprendizaje bidireccional, lo que permite analizar simultáneamente el contexto ubicado tanto a la izquierda como a la derecha de cada palabra durante el proceso de entrenamiento. Esta característica representa una diferencia importante con respecto a modelos anteriores, que procesaban el texto únicamente en una dirección o combinaban representaciones obtenidas de manera independiente [11].

El entrenamiento de BERT se fundamenta en dos estrategias principales: Masked Language Modeling (MLM) y Next Sentence Prediction (NSP). En la primera, el

modelo aprende a predecir palabras que han sido ocultadas dentro de una oración utilizando la información contextual disponible. En la segunda, se entrena para determinar si una oración continúa de manera lógica a otra, favoreciendo la comprensión de las relaciones existentes entre diferentes segmentos de texto. Estas tareas permiten que el modelo adquiera representaciones lingüísticas generales que posteriormente pueden adaptarse a diversas aplicaciones mediante procesos de ajuste fino (*fine-tuning*) [11].

Una de las principales fortalezas de BERT es su capacidad para generar representaciones contextuales dinámicas, donde el significado de una palabra varía de acuerdo con el contexto en el que aparece. Esto permite resolver con mayor precisión fenómenos lingüísticos como la polisemia y la ambigüedad semántica, aspectos que representaban un desafío para modelos basados en representaciones estáticas de palabras. Como consecuencia, BERT ha demostrado un desempeño sobresaliente en tareas como clasificación de textos, análisis de sentimientos, reconocimiento de entidades nombradas, respuesta a preguntas y otras aplicaciones del Procesamiento del Lenguaje Natural [58].

La publicación de BERT marcó un punto de inflexión en la investigación del Procesamiento del Lenguaje Natural y dio origen a múltiples arquitecturas derivadas que buscaron mejorar aspectos relacionados con el rendimiento, la eficiencia computacional y las estrategias de entrenamiento. Modelos como RoBERTa y DistilBERT surgieron a partir de esta arquitectura, incorporando modificaciones orientadas a optimizar su desempeño o reducir sus requerimientos computacionales, manteniendo los principios fundamentales del aprendizaje contextual introducidos por BERT [55].

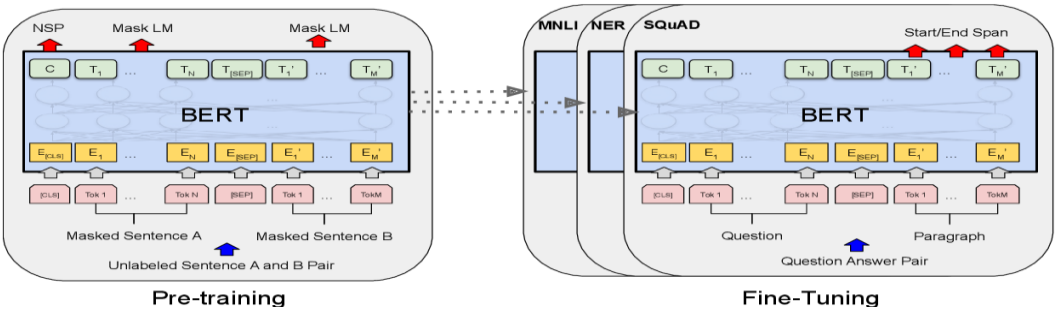


Figura 3. Arquitectura general de BERT basada en el codificador Transformer.[11]

2.2.3.4 RoBERTa

RoBERTa (*Robustly Optimized BERT Pretraining Approach*) es un modelo de lenguaje desarrollado por Liu et al. en 2019 como una versión optimizada de BERT. Su propósito principal consiste en mejorar el rendimiento del modelo mediante ajustes en la estrategia de entrenamiento, sin modificar la arquitectura fundamental basada en el codificador (*encoder*) del Transformer. Los autores demostraron que, al optimizar el proceso de preentrenamiento y emplear una mayor cantidad de datos, era posible obtener mejores resultados en diversas tareas de Procesamiento del Lenguaje Natural, evidenciando que el desempeño de BERT podía incrementarse sin introducir cambios estructurales en su diseño [12].

Entre las principales mejoras incorporadas por RoBERTa se encuentra la eliminación de la tarea Next Sentence Prediction (NSP) durante el preentrenamiento, al considerar que esta no aportaba beneficios significativos al aprendizaje del modelo. Además, se implementó una estrategia de enmascaramiento dinámico (*dynamic masking*), en la cual las palabras ocultas cambian en cada época de entrenamiento, permitiendo que el modelo aprenda representaciones más diversas. Estas modificaciones, junto con el incremento del tamaño de los lotes de entrenamiento y del volumen de datos utilizados, contribuyeron a mejorar la capacidad de generalización del modelo [12].

Diversas evaluaciones han demostrado que RoBERTa supera el rendimiento de BERT en múltiples tareas relacionadas con comprensión del lenguaje, clasificación de textos, inferencia lingüística y respuesta a preguntas. Estos resultados posicionan

a RoBERTa como una de las arquitecturas más utilizadas dentro del Procesamiento del Lenguaje Natural, especialmente en aplicaciones donde se requiere una mayor precisión sin modificar la estructura básica del Transformer. Su desarrollo evidenció que la optimización del proceso de entrenamiento puede ser tan importante como la creación de nuevas arquitecturas [58].

La propuesta de RoBERTa impulsó nuevas investigaciones orientadas a mejorar la eficiencia de los modelos preentrenados. Mientras este modelo priorizó el incremento del rendimiento mediante estrategias de entrenamiento más robustas, otros trabajos se enfocaron en reducir los requerimientos computacionales sin afectar significativamente la precisión. Entre ellos destaca DistilBERT, una arquitectura diseñada para conservar gran parte del desempeño de BERT utilizando un modelo más ligero y eficiente [54].

2.2.3.5 DistilBERT

DistilBERT (Distilled Bidirectional Encoder Representations from Transformers) es un modelo de lenguaje desarrollado por Sanh et al. en 2019 con el objetivo de reducir el tamaño y los requerimientos computacionales de BERT, manteniendo un nivel de desempeño competitivo en diversas tareas de Procesamiento del Lenguaje Natural. Para alcanzar este propósito, los autores emplearon la técnica de destilación del conocimiento (Knowledge Distillation), mediante la cual un modelo de menor tamaño aprende a reproducir el comportamiento de un modelo previamente entrenado de mayor complejidad. Como resultado, DistilBERT conserva gran parte de la capacidad de comprensión del lenguaje de BERT, utilizando menos parámetros y demandando menores recursos de procesamiento [13].

La arquitectura de DistilBERT mantiene los principios fundamentales del codificador (encoder) del Transformer, pero reduce aproximadamente a la mitad el número de capas presentes en BERT. Durante el proceso de entrenamiento, el modelo estudiante aprende a partir de las representaciones generadas por el modelo original, lo que permite preservar el conocimiento adquirido durante el preentrenamiento. Esta estrategia disminuye el tiempo de entrenamiento y de

inferencia, además de reducir el consumo de memoria, convirtiendo a DistilBERT en una alternativa adecuada para entornos con recursos computacionales limitados [13].

Diversos estudios han demostrado que DistilBERT conserva un alto porcentaje del rendimiento obtenido por BERT, a pesar de contar con una arquitectura más compacta. Esta relación entre precisión y eficiencia ha favorecido su incorporación en aplicaciones donde la velocidad de procesamiento y el consumo de recursos constituyen factores determinantes, como sistemas en dispositivos móviles, servicios en la nube y aplicaciones que requieren respuestas en tiempo real. Su desarrollo evidencia que es posible optimizar el rendimiento computacional sin comprometer significativamente la calidad de las representaciones lingüísticas [58].

El surgimiento de modelos como DistilBERT impulsó el interés por desarrollar arquitecturas cada vez más eficientes y versátiles. Posteriormente, esta evolución dio paso a modelos de lenguaje capaces de seguir instrucciones y resolver múltiples tareas utilizando un número reducido de ejemplos, entre los que destaca FLAN-T5, el cual amplía las capacidades de generalización mediante estrategias de aprendizaje basadas en instrucciones (instruction tuning) [54].

2.2.3.6 FLAN-T5

FLAN-T5 (Fine-tuned Language Net Text-to-Text Transfer Transformer 5) es un modelo de lenguaje desarrollado por Chung et al. (2022) a partir de la arquitectura T5 (Text-to-Text Transfer Transformer), propuesta originalmente por Raffel et al en el 2020. A diferencia de modelos como BERT, RoBERTa y DistilBERT, que están orientados principalmente a la comprensión del lenguaje, FLAN-T5 adopta un enfoque texto a texto, en el cual tanto la entrada como la salida se representan en formato textual. Esta característica permite abordar una amplia variedad de tareas mediante una misma arquitectura, transformando cada problema en una instrucción seguida de una respuesta generada por el modelo [21], [59].

La arquitectura de FLAN-T5 se basa en un esquema encoder-decoder derivado del Transformer. En este diseño, el encoder procesa y comprende la información de entrada, mientras que el decoder genera la secuencia de salida de manera

autoregresiva, considerando el contexto previamente aprendido. Gracias a esta estructura, el modelo puede realizar tareas como traducción automática, resumen de textos, respuesta a preguntas, clasificación, generación de contenido y otras aplicaciones del Procesamiento del Lenguaje Natural utilizando un mismo mecanismo de aprendizaje [59].

La principal contribución de FLAN-T5 es la incorporación del Instruction Tuning, estrategia mediante la cual el modelo es entrenado utilizando un amplio conjunto de instrucciones expresadas en lenguaje natural correspondientes a múltiples tareas. Este proceso favorece que el modelo aprenda patrones generales de razonamiento y desarrolle una mayor capacidad para interpretar instrucciones previamente no observadas, mejorando así su desempeño en escenarios donde la disponibilidad de datos etiquetados es limitada [21].

Debido a estas características, FLAN-T5 ha demostrado un desempeño sobresaliente en estrategias de Few-Shot Learning, donde el modelo es capaz de resolver tareas a partir de un número reducido de ejemplos incluidos dentro de la propia instrucción. Esta capacidad de generalización ha convertido a FLAN-T5 en una de las arquitecturas más utilizadas para resolver problemas de Procesamiento del Lenguaje Natural con disponibilidad limitada de datos etiquetados, consolidándose como un referente dentro de los modelos de lenguaje de gran escala basados en instrucciones [60].

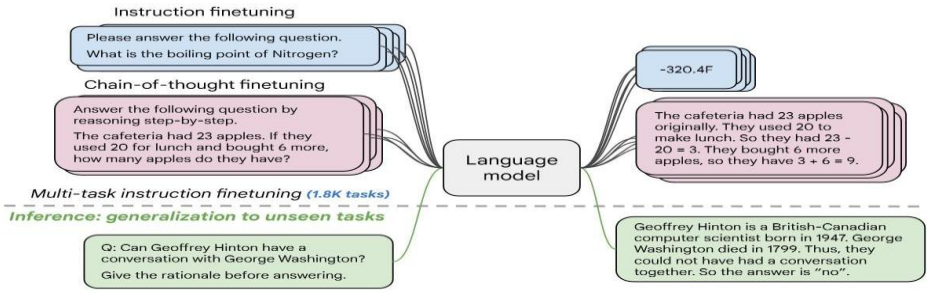


Figura 4. Proceso de ajuste mediante instrucciones (Instruction Tuning) y capacidad de generalización del modelo FLAN-T5 [59].

2.2.4. Técnicas de aprendizaje

Las técnicas de aprendizaje constituyen un conjunto de estrategias empleadas en el aprendizaje automático para entrenar modelos capaces de resolver tareas

específicas a partir de los datos disponibles. La selección de una técnica depende de diversos factores, entre ellos la cantidad de ejemplos etiquetados, la complejidad del problema y la capacidad de generalización requerida. En el ámbito del Procesamiento del Lenguaje Natural, estas estrategias han evolucionado desde métodos tradicionales de aprendizaje supervisado hasta enfoques que aprovechan modelos preentrenados para disminuir la dependencia de grandes volúmenes de datos etiquetados [25].

Con el desarrollo de los modelos basados en Transformer surgieron nuevas estrategias que permiten reutilizar el conocimiento adquirido durante el preentrenamiento para resolver diferentes tareas. Entre ellas se encuentran el ajuste fino (fine-tuning), mediante el cual un modelo preentrenado se adapta a una tarea específica utilizando datos etiquetados, y los enfoques basados en un número reducido de ejemplos, como Zero-Shot Learning, One-Shot Learning y Few-Shot Learning, que aprovechan la capacidad de generalización de los modelos para abordar nuevas tareas con información limitada [61].

En la actualidad, las técnicas de aprendizaje desempeñan un papel fundamental en el desarrollo de aplicaciones de inteligencia artificial, ya que permiten seleccionar el enfoque más adecuado de acuerdo con las características del problema y la disponibilidad de los datos. Su evolución ha impulsado el desarrollo de modelos más flexibles y eficientes, capaces de mantener un alto desempeño incluso en escenarios con recursos limitados para el entrenamiento. En este contexto, resulta importante comprender los fundamentos de la clasificación multietiqueta, las diferentes modalidades del aprendizaje N-Shot y el Few-Shot Learning, debido a su amplia aplicación en tareas modernas de Procesamiento del Lenguaje Natural [62].

2.2.4.1 Clasificación multietiqueta

La clasificación multietiqueta es una técnica de aprendizaje automático que permite asignar simultáneamente dos o más etiquetas a una misma instancia. A diferencia de la clasificación de etiqueta única, donde cada observación pertenece exclusivamente a una categoría, este enfoque reconoce que una instancia puede presentar múltiples características de manera concurrente. Tsoumakas et al.

destacan que este paradigma constituye una alternativa adecuada para representar problemas complejos en los que las categorías no son mutuamente excluyentes [25].

En el ámbito del Procesamiento del Lenguaje Natural, la clasificación multietiqueta ha adquirido una gran relevancia debido a que un mismo texto puede expresar simultáneamente diferentes emociones, temas, intenciones o características semánticas. Esta capacidad ha favorecido su aplicación en tareas como el análisis de sentimientos, la detección de emociones, la clasificación temática, el reconocimiento de entidades y la identificación de rasgos de personalidad, donde una única categoría resulta insuficiente para describir completamente el contenido de un documento [61].

A diferencia de la clasificación multiclase, en la cual las categorías son mutuamente excluyentes y cada instancia solo puede pertenecer a una de ellas, la clasificación multietiqueta permite asignar varias categorías de forma independiente a una misma observación. Esta característica proporciona una representación más flexible de los datos y permite modelar de manera más precisa problemas donde existe coexistencia entre diferentes atributos o clases [62].

El crecimiento de los modelos de aprendizaje profundo y de las arquitecturas basadas en Transformer ha impulsado el uso de la clasificación multietiqueta en aplicaciones modernas de inteligencia artificial. La capacidad de estos modelos para aprender representaciones contextuales ha contribuido a mejorar el desempeño en problemas donde múltiples etiquetas pueden aparecer simultáneamente, consolidando este enfoque como una estrategia ampliamente utilizada en el análisis automatizado de textos [61], [62].

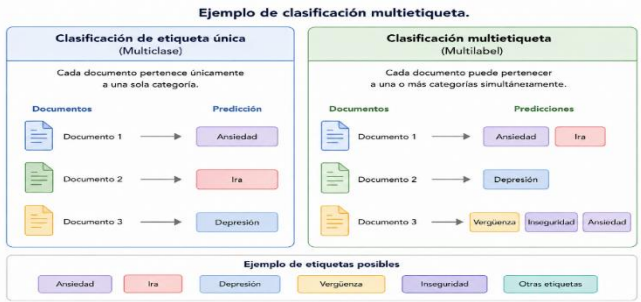


Figura 5. Ejemplo de clasificación multietiqueta.

2.2.4.2 Aprendizaje N-Shot

El aprendizaje N-Shot es un enfoque de aprendizaje automático diseñado para desarrollar modelos capaces de resolver nuevas tareas utilizando una cantidad limitada de ejemplos durante el proceso de adaptación. A diferencia de los métodos tradicionales, que requieren grandes volúmenes de datos etiquetados para alcanzar un buen desempeño, este paradigma busca aprovechar el conocimiento adquirido durante el preentrenamiento para generalizar hacia tareas que disponen de pocos o ningún ejemplo específico. Wang et al. destacan que este enfoque ha cobrado gran importancia debido a la creciente necesidad de desarrollar modelos eficientes en escenarios con disponibilidad limitada de datos [63].

El desarrollo del aprendizaje N-Shot ha estado estrechamente relacionado con la evolución de los modelos de lenguaje basados en Transformer. Gracias a su entrenamiento previo sobre grandes colecciones de texto, estos modelos adquieren representaciones lingüísticas que pueden reutilizarse para resolver diferentes tareas con una mínima cantidad de ejemplos adicionales. Esta capacidad ha permitido reducir significativamente los costos asociados a la recopilación y etiquetado de datos, favoreciendo el desarrollo de aplicaciones en diversos dominios del Procesamiento del Lenguaje Natural [64].

Dentro de este paradigma existen diferentes modalidades que se distinguen por la cantidad de ejemplos proporcionados al modelo durante la fase de inferencia o adaptación. Entre ellas se encuentran el Zero-Shot Learning, que no utiliza ejemplos específicos de la tarea; el One-Shot Learning, que emplea un único ejemplo; y el Few-Shot Learning, que incorpora un número reducido de ejemplos para orientar el comportamiento del modelo. Estas modalidades representan distintos niveles de información disponible y constituyen estrategias ampliamente utilizadas en modelos de lenguaje de gran escala [64], [65].

En la actualidad, el aprendizaje N-Shot se ha consolidado como una de las estrategias más relevantes en inteligencia artificial debido a su capacidad para afrontar tareas con recursos limitados sin comprometer significativamente el rendimiento de los modelos. Su aplicación ha impulsado el desarrollo de soluciones más flexibles y eficientes en áreas como la clasificación de textos, la traducción

automática, la generación de contenido, la respuesta a preguntas y otras tareas del Procesamiento del Lenguaje Natural [64], [89].

2.2.4.3 Modalidades de N-Shot

Las modalidades del aprendizaje N-Shot se diferencian principalmente por la cantidad de ejemplos proporcionados al modelo para resolver una tarea específica. Este enfoque busca aprovechar el conocimiento adquirido durante el preentrenamiento para responder a nuevos problemas con distintos niveles de información disponible, permitiendo adaptar el comportamiento del modelo sin necesidad de realizar un entrenamiento completo [63].

La primera modalidad corresponde al Zero-Shot Learning (ZSL), en la cual el modelo no recibe ejemplos específicos de la tarea durante la inferencia. En este caso, la respuesta se genera únicamente a partir de la instrucción o descripción proporcionada por el usuario y del conocimiento previamente adquirido durante el preentrenamiento. Esta estrategia resulta útil cuando no existen datos etiquetados para una tarea determinada o cuando la obtención de ejemplos es limitada [64].

Por su parte, el One-Shot Learning (OSL) incorpora un único ejemplo representativo de la tarea antes de realizar la predicción. Este ejemplo actúa como una referencia que orienta al modelo sobre el formato esperado de la respuesta, permitiéndole adaptar su comportamiento con una cantidad mínima de información. Aunque ofrece un mayor contexto que el enfoque Zero-Shot, su capacidad de generalización continúa dependiendo en gran medida del conocimiento adquirido durante el preentrenamiento [63].

Finalmente, el Few-Shot Learning (FSL) proporciona al modelo un número reducido de ejemplos antes de la inferencia, con el propósito de mostrar el patrón que debe seguir para resolver la tarea. Al disponer de varios ejemplos representativos, el modelo puede comprender con mayor precisión el contexto y el formato esperado de la respuesta, logrando generalmente un mejor desempeño que las modalidades anteriores en tareas complejas de Procesamiento del Lenguaje Natural [33].

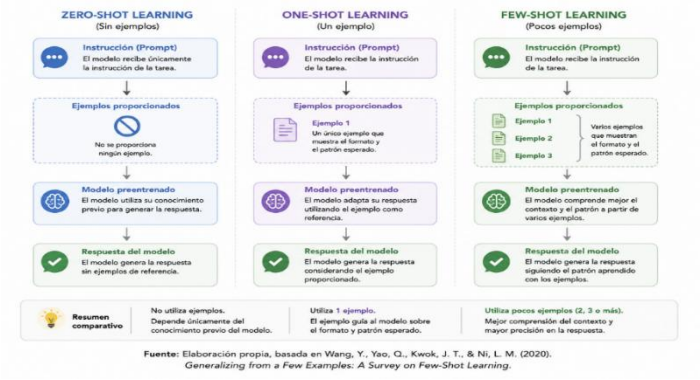


Figura 6. Comparación de las modalidades del aprendizaje N-Shot [63].

2.2.4.4 Few-shot Learning

El Few-Shot Learning (FSL) es una modalidad del aprendizaje N-Shot que permite a un modelo resolver una tarea utilizando un número reducido de ejemplos como referencia. A diferencia de los enfoques tradicionales de aprendizaje supervisado, que requieren grandes conjuntos de datos etiquetados para alcanzar un buen desempeño, el Few-Shot Learning aprovecha el conocimiento adquirido durante el preentrenamiento para adaptarse a nuevas tareas mediante unos pocos ejemplos representativos. Wang et al. señalan que este paradigma constituye una alternativa eficaz cuando la disponibilidad de datos es limitada o el proceso de etiquetado resulta costoso [63].

El funcionamiento del Few-Shot Learning se basa en proporcionar al modelo una instrucción acompañada de un pequeño conjunto de ejemplos que ilustran la tarea que debe realizar. Estos ejemplos permiten que el modelo identifique patrones, relaciones y el formato esperado de la respuesta, favoreciendo una mejor comprensión del contexto sin necesidad de un proceso adicional de entrenamiento. Gracias a este mecanismo, los modelos de lenguaje de gran escala pueden generalizar su conocimiento hacia tareas nuevas con un número reducido de muestras [64].

Una de las principales ventajas del Few-Shot Learning es su capacidad para disminuir la dependencia de grandes volúmenes de datos etiquetados, reduciendo el tiempo y los recursos necesarios para desarrollar soluciones basadas en inteligencia artificial. Asimismo, este enfoque facilita la adaptación de modelos preentrenados a diferentes dominios y tareas, manteniendo un desempeño

competitivo incluso cuando la información disponible es escasa. Estas características han favorecido su adopción en aplicaciones como la clasificación de textos, la traducción automática, la generación de contenido y la respuesta a preguntas [21].

El crecimiento de los modelos de lenguaje basados en Transformer ha impulsado la consolidación del Few-Shot Learning como una de las estrategias más utilizadas en el Procesamiento del Lenguaje Natural. Modelos como FLAN-T5 incorporan el aprendizaje mediante instrucciones (instruction tuning), lo que mejora su capacidad para interpretar tareas y generar respuestas coherentes utilizando únicamente unos pocos ejemplos. Como resultado, el Few-Shot Learning se ha convertido en un enfoque ampliamente empleado para resolver problemas donde la disponibilidad de datos etiquetados es reducida [64], [21].

2.2.5. Métricas de evaluación

Las métricas de evaluación constituyen un conjunto de indicadores utilizados para medir el desempeño de los modelos de aprendizaje automático durante las etapas de validación y prueba. Su principal propósito es cuantificar la capacidad del modelo para realizar predicciones correctas y proporcionar información objetiva sobre su nivel de precisión, confiabilidad y capacidad de generalización. En el ámbito del Procesamiento del Lenguaje Natural, estas métricas permiten comparar diferentes modelos y determinar cuál ofrece un mejor rendimiento para una tarea específica [66].

La selección de las métricas de evaluación depende de las características del problema de aprendizaje y del tipo de datos utilizados. En tareas de clasificación, especialmente aquellas donde pueden existir múltiples categorías asociadas a una misma instancia, resulta necesario emplear métricas que permitan evaluar distintos aspectos del comportamiento del modelo, como la proporción de predicciones correctas, la capacidad para identificar correctamente las clases de interés y el equilibrio entre ambas. De esta manera, es posible obtener una evaluación más completa que la proporcionada por una única medida de desempeño [30].

En los últimos años, el desarrollo de modelos basados en aprendizaje profundo ha incrementado la importancia de utilizar métricas de evaluación adecuadas para interpretar de forma objetiva los resultados obtenidos. En problemas de clasificación multietiqueta, métricas como Precisión (Precision), Exhaustividad (Recall), F1-Score y Hamming Loss se han consolidado como algunas de las más empleadas debido a que ofrecen una visión integral del rendimiento del modelo desde diferentes perspectivas. Estas métricas facilitan la comparación entre modelos y contribuyen a una evaluación más confiable de su capacidad predictiva [61].

2.2.5.1 Precisión

La Precisión (Precision) es una métrica de evaluación utilizada para medir la proporción de predicciones positivas realizadas correctamente por un modelo con respecto al total de predicciones positivas generadas. Su objetivo es determinar qué tan confiables son las predicciones clasificadas como positivas, por lo que constituye un indicador ampliamente empleado en problemas de clasificación, especialmente cuando los errores por falsos positivos pueden afectar significativamente la calidad de los resultados. Powers destaca que esta métrica permite evaluar la exactitud de las predicciones positivas emitidas por un clasificador [66].

La precisión se calcula mediante la relación entre el número de verdaderos positivos (True Positives, TP) y la suma de los verdaderos positivos junto con los falsos positivos (False Positives, FP). Un valor elevado de esta métrica indica que la mayoría de las instancias clasificadas como positivas corresponden realmente a dicha categoría, reflejando un bajo número de predicciones positivas incorrectas [30].

$$\text{Precisión} = \frac{TP}{TP + FP}$$

Donde:

- TP (True Positives): número de instancias positivas correctamente clasificadas.

- FP (False Positives): número de instancias clasificadas como positivas cuando en realidad pertenecen a otra categoría.

En problemas de clasificación multietiqueta, la precisión adquiere una importancia particular debido a que cada instancia puede estar asociada simultáneamente a varias etiquetas. En este contexto, la métrica permite evaluar la confiabilidad de las etiquetas predichas por el modelo, proporcionando información sobre la proporción de etiquetas asignadas correctamente respecto al total de etiquetas positivas predichas. Por esta razón, la precisión constituye uno de los indicadores más utilizados para comparar el desempeño de modelos de aprendizaje profundo aplicados al Procesamiento del Lenguaje Natural [61].

Figura 7. Representación conceptual de la métrica Precisión.

2.2.5.2 Exhaustividad (Recall)

La Exhaustividad (Recall) es una métrica de evaluación utilizada para medir la capacidad de un modelo para identificar con corrección las instancias que realmente pertenecen a una clase positiva. A diferencia de la precisión, que evalúa la confiabilidad de las predicciones positivas realizadas por el modelo, el Recall determina qué proporción de los casos positivos existentes fue detectada correctamente. Powers señala que esta métrica es especialmente importante en problemas donde la omisión de instancias positivas puede tener un impacto significativo sobre el desempeño del sistema [66].

El Recall se calcula mediante la relación entre el número de verdaderos positivos (True Positives, TP) y la suma de los verdaderos positivos junto con los falsos negativos (False Negatives, FN). Un valor elevado indica que el modelo logra

identificar la mayoría de las instancias positivas presentes en los datos, reduciendo la cantidad de casos que deberían haber sido detectados, pero fueron clasificados incorrectamente como negativos [30].

$$\text{Recall} = \frac{TP}{TP + FN}$$

Donde:

- TP (True Positives): número de instancias positivas correctamente clasificadas.
- FN (False Negatives): número de instancias positivas que el modelo clasificó incorrectamente como negativas.

En problemas de clasificación multietiqueta, el Recall permite evaluar la capacidad del modelo para recuperar las etiquetas que realmente corresponden a cada instancia. Esta métrica resulta especialmente útil cuando el objetivo es minimizar la pérdida de información relevante, ya que penaliza las etiquetas positivas que no fueron identificadas durante el proceso de clasificación. Por esta razón, constituye una de las métricas más empleadas para evaluar modelos de aprendizaje profundo aplicados al Procesamiento del Lenguaje Natural [61].

2.2.5.3 F1-Score

El F1-Score es una métrica de evaluación que combina la Precisión (Precisión) y la Exhaustividad (Recall) en un único indicador, permitiendo valorar de manera equilibrada el desempeño de un modelo de clasificación. A diferencia de utilizar ambas métricas por separado, el F1-Score considera simultáneamente la capacidad del modelo para realizar predicciones positivas correctas y para identificar la mayor cantidad posible de instancias positivas. Sokolova y Lapalme señalan que esta métrica resulta especialmente útil cuando existe un desbalance entre las clases o cuando es necesario encontrar un equilibrio entre precisión y exhaustividad [30].

El F1-Score se calcula mediante la media armónica entre la Precisión y el Recall, lo que implica que un valor elevado solo se obtiene cuando ambas métricas presentan un buen desempeño. Si una de ellas es considerablemente menor que la otra, el valor del F1-Score disminuye, reflejando un desequilibrio en la capacidad

predictiva del modelo. Esta característica convierte al F1-Score en una medida más representativa que la exactitud en numerosos problemas de clasificación [66].

$$F1 = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precisión} + \text{Recall}}$$

Donde:

- Precisión: proporción de predicciones positivas correctamente realizadas respecto al total de predicciones positivas.
- Recall: proporción de instancias positivas correctamente identificadas respecto al total de instancias positivas reales.

En tareas de clasificación multietiqueta, el F1-Score es ampliamente utilizado debido a que proporciona una evaluación global del desempeño del modelo al considerar simultáneamente la precisión y la capacidad de recuperación de las etiquetas. Esta métrica facilita la comparación entre diferentes modelos de aprendizaje profundo y constituye uno de los indicadores más empleados para evaluar sistemas de Procesamiento del Lenguaje Natural, especialmente cuando las clases presentan distribuciones desbalanceadas [61].

2.2.5.4 Hamming Loss

El Hamming Loss es una métrica de evaluación utilizada principalmente en problemas de clasificación multietiqueta, cuyo propósito es medir la proporción de etiquetas clasificadas incorrectamente respecto al número total de etiquetas evaluadas. A diferencia de métricas como la Precisión, el Recall y el F1-Score, que se centran en el desempeño de las predicciones positivas, el Hamming Loss cuantifica los errores cometidos por el modelo considerando tanto las etiquetas positivas como las negativas. Zhang y Zhou destacan que esta métrica proporciona una evaluación adecuada en problemas donde cada instancia puede estar asociada simultáneamente a múltiples etiquetas [61]..

El Hamming Loss se calcula como la proporción de etiquetas predichas incorrectamente con respecto al total de etiquetas evaluadas. Su valor oscila entre 0 y 1, donde un resultado cercano a 0 indica un mejor desempeño del modelo al representar una menor cantidad de errores de clasificación. Por el contrario, valores

próximos a 1 reflejan un mayor número de predicciones incorrectas y, por tanto, un menor rendimiento del sistema [62].

$$\text{Hamming Loss} = \frac{1}{N \times L} \sum_{i=1}^{N} \sum_{j=1}^{L} I\left(y_{ij} \neq \hat{y}_{ij}\right)$$

Donde:

- N: número total de instancias evaluadas.
- L: número de etiquetas consideradas.
- y_{ij}: valor real de la etiqueta *j* para la instancia *i*.
- $\hat{y}_{ij}$: valor predicho por el modelo para la etiqueta *j* de la instancia *i*.
- $I(\cdot)$: función indicadora, que toma el valor 1 cuando existe un error de clasificación y 0 cuando la predicción es correcta.

En aplicaciones de Procesamiento del Lenguaje Natural, el Hamming Loss es ampliamente utilizado para evaluar modelos de clasificación multietiqueta debido a que considera los errores cometidos en cada etiqueta de manera independiente. Esta característica permite obtener una medida más detallada del comportamiento del modelo, especialmente cuando una misma instancia puede pertenecer simultáneamente a varias categorías. Por ello, esta métrica constituye un complemento importante de la Precisión, el Recall y el F1-Score en la evaluación integral del desempeño de modelos de aprendizaje profundo [25].

2.2.6. Entorno de desarrollo

El entorno de desarrollo comprende el conjunto de herramientas, bibliotecas y plataformas utilizadas para implementar, entrenar y evaluar modelos de aprendizaje automático. La selección de estos recursos influye directamente en la eficiencia del proceso de desarrollo, ya que permite optimizar el procesamiento de datos, la construcción de modelos y la ejecución de experimentos. En proyectos relacionados con el Procesamiento del Lenguaje Natural, disponer de un entorno adecuado

facilita la integración de diferentes tecnologías especializadas para cada etapa del flujo de trabajo [67].

El crecimiento de las aplicaciones basadas en inteligencia artificial ha impulsado el desarrollo de plataformas y bibliotecas orientadas al análisis de datos y al entrenamiento de modelos de aprendizaje profundo. Herramientas como Google Colab y lenguajes de programación como Python permiten desarrollar soluciones de manera colaborativa y aprovechar recursos computacionales de alto rendimiento, mientras que bibliotecas especializadas simplifican tareas relacionadas con el procesamiento, análisis y visualización de datos [68].

En el ámbito del Procesamiento del Lenguaje Natural, el uso conjunto de bibliotecas como Pandas, NumPy, NLTK, spaCy, Matplotlib y Seaborn proporciona un ecosistema de trabajo que facilita desde la preparación de los datos hasta la representación gráfica de los resultados obtenidos. La integración de estas herramientas ha favorecido el desarrollo de soluciones más eficientes, reproducibles y escalables para aplicaciones basadas en aprendizaje profundo [69].

2.2.6.1 Google Colab

Google Colab es una plataforma de computación en la nube desarrollada por Google que permite ejecutar código en Python directamente desde un navegador web sin necesidad de realizar instalaciones locales. Basada en los cuadernos interactivos de Jupyter Notebook, esta herramienta facilita el desarrollo de proyectos relacionados con ciencia de datos, aprendizaje automático e inteligencia artificial, proporcionando un entorno accesible para la creación, ejecución y documentación de experimentos [70].

Una de las principales ventajas de Google Colab es el acceso gratuito a recursos computacionales de alto rendimiento, como unidades de procesamiento gráfico (GPU) y unidades de procesamiento tensorial (TPU), las cuales aceleran significativamente el entrenamiento de modelos de aprendizaje profundo. Además, su integración con Google Drive permite almacenar y compartir archivos de forma sencilla, favoreciendo el trabajo colaborativo y la reproducibilidad de los experimentos [71].

Gracias a estas características, Google Colab se ha convertido en una de las plataformas más utilizadas en investigaciones relacionadas con el Procesamiento del Lenguaje Natural y el aprendizaje profundo. Su compatibilidad con bibliotecas como TensorFlow, PyTorch, Scikit-learn y Hugging Face Transformers facilita el desarrollo de modelos de inteligencia artificial, permitiendo integrar en un mismo entorno las diferentes etapas del procesamiento, entrenamiento y evaluación de los modelos [72].

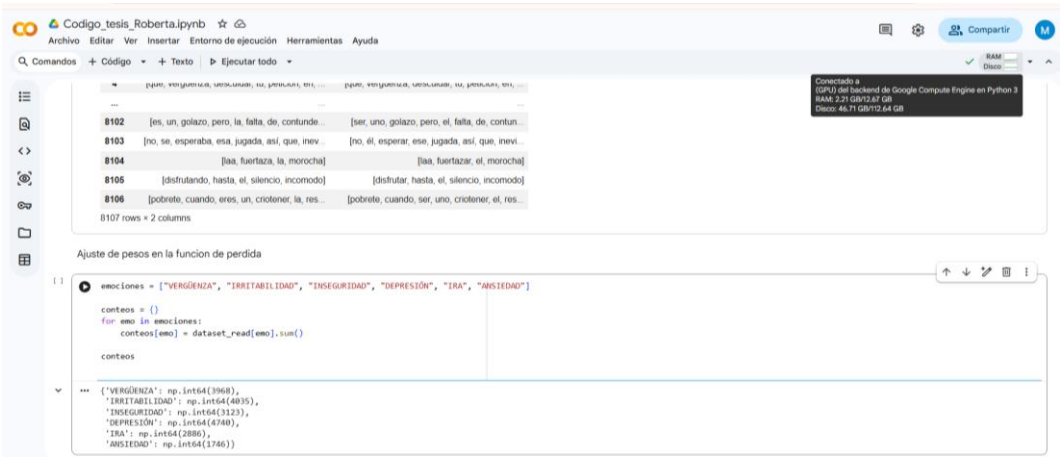


Figura 8. Entorno de desarrollo en Google Colab. Elaboración propia.

2.2.6.2 Python

Python es un lenguaje de programación de alto nivel, interpretado y de propósito general, ampliamente utilizado en el desarrollo de aplicaciones científicas, análisis de datos, aprendizaje automático e inteligencia artificial. Su sintaxis sencilla y legible facilita el desarrollo de programas complejos, permitiendo a investigadores y desarrolladores implementar soluciones de manera eficiente. Además, dispone de un amplio ecosistema de bibliotecas especializadas que respaldan tareas relacionadas con el Procesamiento del Lenguaje Natural y el aprendizaje profundo [73].

Una de las principales características de Python es su portabilidad y compatibilidad con múltiples sistemas operativos, lo que permite ejecutar un mismo programa en diferentes entornos sin realizar modificaciones significativas. Asimismo, su naturaleza de código abierto ha impulsado una amplia comunidad de desarrolladores que contribuye continuamente al desarrollo de nuevas herramientas

y bibliotecas, fortaleciendo su adopción en proyectos científicos y tecnológicos [74].

En el ámbito de la inteligencia artificial, Python se ha consolidado como uno de los lenguajes de programación más utilizados debido a su integración con bibliotecas como NumPy, Pandas, NLTK, spaCy, PyTorch y Hugging Face Transformers. Estas herramientas permiten desarrollar aplicaciones orientadas al análisis de datos, entrenamiento de modelos de aprendizaje profundo y procesamiento de lenguaje natural, convirtiendo a Python en una plataforma fundamental para investigaciones y aplicaciones basadas en inteligencia artificial [67].

2.2.6.3 Pandas

Pandas es una biblioteca de código abierto para Python especializada en la manipulación, organización y análisis de datos estructurados. Desarrollada por Wes McKinney, proporciona estructuras de datos eficientes, como Series y DataFrame, que permiten almacenar, transformar y analizar grandes volúmenes de información de manera sencilla. Gracias a su flexibilidad y facilidad de uso, Pandas se ha consolidado como una de las herramientas más utilizadas en proyectos de ciencia de datos, aprendizaje automático e inteligencia artificial [69].

Entre las principales funcionalidades de Pandas se encuentran la lectura y escritura de datos en diversos formatos, como CSV, Excel, JSON y bases de datos, así como la limpieza, filtrado, combinación y transformación de conjuntos de datos. Estas capacidades facilitan el tratamiento de información antes de su procesamiento, permitiendo detectar valores faltantes, eliminar registros duplicados y reorganizar la estructura de los datos para su posterior análisis [75].

En proyectos de Procesamiento del Lenguaje Natural, Pandas desempeña un papel fundamental en la preparación y gestión de conjuntos de datos textuales. Su integración con bibliotecas como NumPy, Scikit-learn y Hugging Face Transformers permite construir flujos de trabajo eficientes para el preprocesamiento, análisis y entrenamiento de modelos de aprendizaje profundo, contribuyendo a una manipulación organizada y reproducible de la información [69].

2.2.6.4 NumPy

NumPy (Numerical Python) es una biblioteca de código abierto para Python especializada en la computación numérica y el procesamiento eficiente de datos multidimensionales. Desarrollada para optimizar operaciones matemáticas y algebraicas, proporciona la estructura de datos ndarray, la cual permite almacenar y manipular grandes volúmenes de información de forma más eficiente que las estructuras nativas del lenguaje. Gracias a su alto rendimiento y versatilidad, NumPy constituye una de las bibliotecas fundamentales para la ciencia de datos, el aprendizaje automático y la inteligencia artificial [76].

Entre las principales funcionalidades de NumPy se encuentran la ejecución de operaciones matemáticas vectorizadas, el manejo de matrices multidimensionales, la generación de datos numéricos y la implementación de funciones estadísticas y algebraicas. Estas características permiten realizar cálculos complejos con un menor consumo de recursos computacionales, favoreciendo el procesamiento eficiente de grandes conjuntos de datos y la preparación de información para diferentes algoritmos de aprendizaje automático [76].

En aplicaciones de Procesamiento del Lenguaje Natural, NumPy desempeña un papel importante en la representación y manipulación de datos numéricos derivados de los textos, como vectores, matrices y tensores utilizados durante el entrenamiento y evaluación de modelos de aprendizaje profundo. Además, su integración con bibliotecas como Pandas, PyTorch, Scikit-learn y Hugging Face Transformers facilita la construcción de flujos de trabajo eficientes para el análisis y procesamiento de información textual [76].

2.2.6.5 NLTK

NLTK (Natural Language Toolkit) es una biblioteca de código abierto desarrollada para Python que proporciona un amplio conjunto de herramientas destinadas al análisis y procesamiento del lenguaje natural. Creada por Bird, Klein y Loper, esta biblioteca facilita la implementación de diversas tareas lingüísticas mediante recursos léxicos, algoritmos y corpus de referencia, convirtiéndose en una de las

herramientas más utilizadas para la enseñanza, investigación y desarrollo de aplicaciones basadas en texto [26].

Entre las funcionalidades que ofrece NLTK se encuentran la tokenización, la eliminación de palabras vacías (stopwords), el análisis léxico, el etiquetado gramatical, la clasificación de textos y el acceso a diferentes corpus lingüísticos. Estas capacidades permiten preparar y transformar los datos textuales antes de su procesamiento, mejorando la calidad de la información utilizada durante las etapas de entrenamiento y evaluación de modelos de aprendizaje automático [77].

En proyectos de Procesamiento del Lenguaje Natural, NLTK constituye una herramienta fundamental para las tareas de preprocesamiento, ya que proporciona métodos eficientes para segmentar textos, normalizar información y eliminar elementos que no aportan valor al análisis. Además, su integración con bibliotecas como Pandas, NumPy, spaCy y Hugging Face Transformers facilita la construcción de flujos de trabajo completos orientados al desarrollo de modelos de inteligencia artificial aplicados al análisis de textos [26].

2.2.6.6 SpaCy

SpaCy es una biblioteca de código abierto para Python especializada en el Procesamiento del Lenguaje Natural, diseñada para ofrecer un alto rendimiento en aplicaciones que requieren el análisis eficiente de grandes volúmenes de texto. Desarrollada por Explosión AI, proporciona modelos lingüísticos entrenados para múltiples idiomas y un conjunto de herramientas optimizadas para realizar tareas de análisis sintáctico, reconocimiento de entidades, lematización y procesamiento de documentos textuales. Su rapidez y precisión la han convertido en una de las bibliotecas más utilizadas en proyectos de inteligencia artificial y minería de textos [78].

Entre las principales funcionalidades de spaCy se encuentran la tokenización, la lematización, el etiquetado gramatical (Part-of-Speech Tagging), el análisis de dependencias sintácticas y el reconocimiento de entidades nombradas (Named Entity Recognition). Estas herramientas permiten transformar el lenguaje natural en representaciones estructuradas que facilitan el análisis automático de la información

textual y la extracción de características relevantes para diferentes modelos de aprendizaje automático [79].

En el ámbito del Procesamiento del Lenguaje Natural, spaCy destaca por su integración con bibliotecas como NumPy, Pandas, PyTorch y Hugging Face Transformers, lo que permite construir flujos de trabajo eficientes para el desarrollo de aplicaciones basadas en aprendizaje profundo. Gracias a su arquitectura optimizada y a la disponibilidad de modelos específicos para distintos idiomas, esta biblioteca se ha consolidado como una herramienta ampliamente utilizada en tareas de clasificación de textos, análisis semántico y comprensión del lenguaje [78].

2.2.6.7 Hugging Face Transformers

Hugging Face Transformers es una biblioteca de código abierto para Python desarrollada con el propósito de facilitar la implementación y utilización de modelos de lenguaje basados en la arquitectura Transformer. Creada por la empresa Hugging Face, esta biblioteca ofrece acceso a una amplia colección de modelos preentrenados para diversas tareas de Procesamiento del Lenguaje Natural, visión por computadora y procesamiento multimodal. Su facilidad de integración con los principales marcos de aprendizaje profundo, como PyTorch y TensorFlow, ha favorecido su adopción tanto en el ámbito académico como en aplicaciones industriales relacionadas con la inteligencia artificial [80].

Una de las principales ventajas de esta biblioteca es la disponibilidad de modelos previamente entrenados que pueden adaptarse a tareas específicas mediante el proceso de ajuste fino (fine-tuning) o utilizarse directamente en escenarios de aprendizaje con pocos ejemplos (Few-Shot Learning). Asimismo, proporciona herramientas para la carga de tokenizadores, administración de modelos, gestión de conjuntos de datos y ejecución eficiente de procesos de entrenamiento e inferencia, reduciendo significativamente el tiempo necesario para desarrollar soluciones basadas en modelos Transformer [81].

Además de simplificar la implementación de arquitecturas avanzadas, Hugging Face Transformers promueve la reproducibilidad de los experimentos al ofrecer modelos estandarizados y documentación ampliamente respaldada por la

comunidad científica. Gracias a estas características, la biblioteca se ha convertido en un referente para el desarrollo de aplicaciones de Procesamiento del Lenguaje Natural, permitiendo utilizar modelos como BERT, RoBERTa, DistilBERT y FLAN-T5 en tareas de clasificación de textos, generación de lenguaje, traducción automática y análisis semántico, entre muchas otras aplicaciones [80].

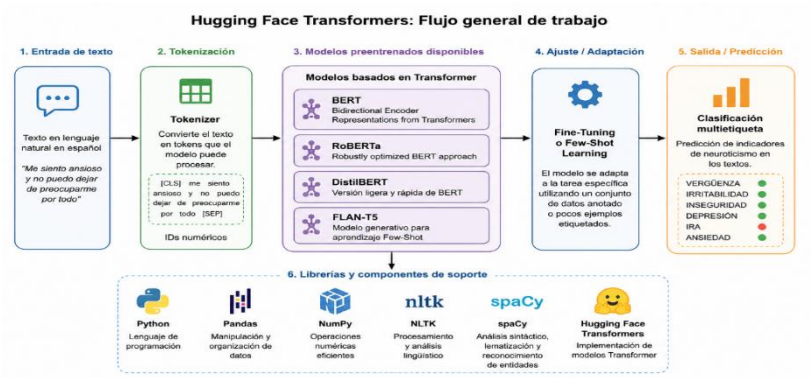


Figura 9. Flujo general de trabajo utilizando la biblioteca Hugging Face Transformers.

2.2.6.8 Matplotlib

Matplotlib es una biblioteca de código abierto para Python especializada en la creación de representaciones gráficas para el análisis y visualización de datos. Desarrollada por John D. Hunter, ofrece una amplia variedad de herramientas para generar gráficos bidimensionales y tridimensionales que facilitan la interpretación de información numérica y estadística. Gracias a su versatilidad y compatibilidad con otras bibliotecas del ecosistema científico de Python, Matplotlib se ha consolidado como una de las herramientas más utilizadas en proyectos de análisis de datos, aprendizaje automático e inteligencia artificial [82].

Entre las principales capacidades de Matplotlib se encuentran la generación de gráficos de líneas, barras, dispersión, histogramas y diagramas de diferentes tipos, así como la personalización de títulos, ejes, leyendas y escalas. Estas funcionalidades permiten representar de manera clara los resultados obtenidos durante el procesamiento y análisis de datos, favoreciendo la comprensión e interpretación de la información por parte de investigadores y usuarios [83].

En el ámbito de la inteligencia artificial y el Procesamiento del Lenguaje Natural, Matplotlib constituye una herramienta de apoyo para la visualización del comportamiento de los modelos durante las etapas de entrenamiento y evaluación.

Su integración con bibliotecas como NumPy, Pandas y Seaborn facilita la elaboración de gráficos utilizados para analizar métricas de rendimiento, comparar resultados experimentales y comunicar de forma efectiva los hallazgos obtenidos en investigaciones científicas [82].

2.2.6.9 Seaborn

Seaborn es una biblioteca de visualización de datos para Python desarrollada sobre Matplotlib, diseñada para facilitar la creación de gráficos estadísticos con una apariencia clara y profesional. Creada por Michael Waskom, proporciona una interfaz de alto nivel que simplifica la representación de conjuntos de datos complejos mediante gráficos informativos y estéticamente consistentes. Gracias a su integración con bibliotecas como Pandas y NumPy, Seaborn se ha convertido en una herramienta ampliamente utilizada en el análisis exploratorio de datos y en la presentación de resultados obtenidos en investigaciones científicas [84].

A diferencia de Matplotlib, que ofrece un mayor nivel de personalización mediante instrucciones más detalladas, Seaborn incorpora funciones especializadas para la visualización de distribuciones, relaciones entre variables y comparaciones estadísticas. Entre sus principales recursos se encuentran los mapas de calor (heatmaps), diagramas de dispersión, gráficos de barras, gráficos de cajas (boxplots) y matrices de correlación, los cuales permiten identificar patrones, tendencias y comportamientos presentes en los datos de manera intuitiva [85].

En el contexto de la inteligencia artificial y el Procesamiento del Lenguaje Natural, Seaborn facilita la representación visual de métricas de evaluación, matrices de confusión y análisis comparativos del rendimiento de distintos modelos. Su capacidad para generar gráficos claros y de alta calidad favorece la interpretación de los resultados experimentales y la comunicación de los hallazgos obtenidos durante el desarrollo de investigaciones basadas en aprendizaje profundo, complementando las funcionalidades de Matplotlib para la elaboración de visualizaciones científicas [85].

2.3. Marco Teórico

El marco teórico constituye el sustento científico de la presente investigación, ya que reúne los principales estudios y aportes relacionados con la detección automática de rasgos de personalidad mediante el análisis de textos. A diferencia del marco conceptual, en esta sección se examina la evolución de las investigaciones, los enfoques metodológicos y los avances alcanzados en el Procesamiento del Lenguaje Natural y el aprendizaje profundo aplicados al análisis de la personalidad. Asimismo, se identifican las principales limitaciones reportadas por la literatura, las cuales justifican el desarrollo de nuevas estrategias para la detección de indicadores de neuroticismo en textos escritos en español.

2.3.1. Evolución del Procesamiento del Lenguaje Natural aplicado al análisis de personalidad

Durante las últimas décadas, el Procesamiento del Lenguaje Natural (PLN) ha experimentado un crecimiento significativo impulsado por el desarrollo de nuevas técnicas de inteligencia artificial y por el incremento de información textual generada en medios digitales. Las primeras investigaciones se centraban en métodos basados en reglas lingüísticas y características elaboradas manualmente; sin embargo, la incorporación de algoritmos de aprendizaje automático permitió mejorar la capacidad de los sistemas para comprender y analizar el lenguaje humano. Posteriormente, la aparición del aprendizaje profundo y de las arquitecturas Transformer marcó un cambio importante en el desempeño de los modelos destinados al análisis de textos [2].

Como resultado de esta evolución, el Procesamiento del Lenguaje Natural ha favorecido el desarrollo de aplicaciones capaces de identificar información semántica, emociones y rasgos de personalidad a partir del lenguaje escrito. Modelos como BERT, RoBERTa y DistilBERT demostraron que las representaciones contextuales permiten comprender con mayor precisión las relaciones entre las palabras y el contexto en el que aparecen, superando las limitaciones de enfoques tradicionales basados únicamente en características léxicas o estadísticas [11], [12], [13].

En la actualidad, el uso de modelos preentrenados y técnicas de aprendizaje con pocos ejemplos ha ampliado significativamente las posibilidades del análisis automático de la personalidad, especialmente en idiomas donde la disponibilidad de datos etiquetados es limitada. Estas estrategias han permitido adaptar modelos de lenguaje a tareas específicas con un menor costo de entrenamiento, favoreciendo el desarrollo de aplicaciones orientadas al análisis del comportamiento humano, la minería de opiniones y la detección de rasgos psicológicos mediante textos digitales [64], [80].

2.3.2. Fundamentación psicolingüística del análisis de personalidad

La psicolingüística sostiene que el lenguaje constituye una manifestación de los procesos cognitivos y emocionales de las personas, razón por la cual el estudio de las expresiones lingüísticas permite identificar patrones relacionados con la personalidad y el comportamiento. Diversas investigaciones han demostrado que la selección de palabras, el uso de pronombres, la estructura de las oraciones y otros elementos del discurso reflejan características psicológicas relativamente estables, convirtiendo al lenguaje en una fuente de información relevante para el análisis automatizado de rasgos de personalidad [86], [87].

El desarrollo de herramientas como *Linguistic Inquiry and Word Count* (LIWC) permitió fortalecer esta línea de investigación al proporcionar recursos para analizar categorías lingüísticas asociadas con variables psicológicas. Gracias a este tipo de herramientas fue posible establecer relaciones entre determinadas características del lenguaje y dimensiones de personalidad, facilitando la construcción de modelos computacionales orientados a la identificación automática de patrones conductuales presentes en textos escritos [6], [88].

Los avances alcanzados en esta área han favorecido la integración entre la psicología y la inteligencia artificial, permitiendo que los modelos actuales no solo analicen la frecuencia de determinadas palabras, sino también el contexto en el que son utilizadas. Como resultado, el análisis psicolingüístico se ha consolidado como uno de los principales fundamentos científicos para el desarrollo de sistemas capaces de inferir rasgos de personalidad mediante técnicas modernas de Procesamiento del Lenguaje Natural [89], [87].

2.3.3. Análisis de la personalidad en textos y su aplicación

El análisis automático de la personalidad ha despertado un creciente interés debido a la expansión de las plataformas digitales y al volumen de información textual que los usuarios generan diariamente. Diversos estudios han demostrado que publicaciones en redes sociales, comentarios, blogs y foros contienen patrones lingüísticos capaces de reflejar rasgos de personalidad, lo que ha impulsado el desarrollo de modelos computacionales orientados a su identificación automática [90], [89].

Los avances alcanzados permitieron extender estas investigaciones hacia diversos ámbitos de aplicación. En psicología contribuyen a complementar procesos de evaluación y estudio del comportamiento humano; en educación permiten analizar patrones de interacción de los estudiantes; mientras que en marketing y ciencias sociales facilitan el conocimiento de preferencias y perfiles de comportamiento a partir de información generada en medios digitales. Estas aplicaciones evidencian el potencial del análisis de personalidad para apoyar procesos de toma de decisiones basados en información textual [90], [87].

En los últimos años, la incorporación de modelos basados en aprendizaje profundo ha permitido mejorar la precisión de estas tareas al considerar el contexto lingüístico de manera más eficiente que los enfoques tradicionales [14]. Como consecuencia, el análisis automático de la personalidad continúa consolidándose como una línea de investigación con importantes aplicaciones en inteligencia artificial y ciencias del comportamiento [80].

2.3.4. Desafíos actuales en la detección automática de rasgos de personalidad

A pesar de los avances alcanzados en el análisis automático de la personalidad, todavía existen diversos desafíos que limitan el desempeño de los modelos de inteligencia artificial. Entre ellos destacan la ambigüedad propia del lenguaje natural, la presencia de ironía, sarcasmo, expresiones coloquiales y diferencias culturales que dificultan la correcta interpretación del significado de los textos. Estas características incrementan la complejidad de los sistemas destinados a identificar rasgos psicológicos mediante información escrita [91], [2].

Otro desafío importante corresponde a la disponibilidad de conjuntos de datos etiquetados, especialmente para idiomas distintos del inglés. En el caso del español, la cantidad de corpus especializados continúa siendo reducida, lo que limita el entrenamiento de modelos complejos y favorece la utilización de estrategias como el aprendizaje con pocos ejemplos (*Few-Shot Learning*) y la reutilización de modelos preentrenados. Estas técnicas permiten aprovechar el conocimiento previamente adquirido para mejorar el desempeño en tareas específicas con una cantidad limitada de datos [64], [18].

2.4.Requerimientos

Para el desarrollo del sistema propuesto se establecieron requerimientos funcionales y no funcionales que definen las capacidades, restricciones y características necesarias para garantizar el correcto funcionamiento de la solución tecnológica. Estos requerimientos permiten delimitar el alcance del sistema y sirven como base durante las etapas de diseño, implementación, pruebas y evaluación del modelo. En el contexto del presente proyecto, los requerimientos se orientan al desarrollo de un sistema inteligente capaz de identificar automáticamente indicadores lingüísticos de neuroticismo en textos en idioma español mediante técnicas de procesamiento de lenguaje natural y modelos basados en arquitecturas Transformer.

La definición de requerimientos constituye una etapa fundamental dentro del proceso de desarrollo, ya que permite especificar de manera estructurada las necesidades funcionales del sistema y los criterios de calidad que debe cumplir durante su ejecución. Además, estos requerimientos facilitan la planificación de los componentes necesarios para el procesamiento de datos, entrenamiento del modelo y visualización de resultados, garantizando una solución escalable, eficiente y alineada con los objetivos de investigación.

2.4.1. Requerimientos Funcionales

Los requerimientos funcionales describen las acciones, procesos y servicios que el sistema debe ejecutar para cumplir con los objetivos planteados en la investigación. Estos requerimientos definen el comportamiento esperado del sistema frente a las interacciones del usuario y establecen las funciones principales necesarias para el

procesamiento y análisis de la información. En el presente proyecto, el sistema debe permitir el ingreso de textos en idioma español, realizar su procesamiento automático, generar representaciones vectoriales contextuales y ejecutar una clasificación multietiqueta orientada a la detección de indicadores de neuroticismo.

Asimismo, el sistema debe presentar los resultados de manera clara y comprensible, permitiendo al usuario interpretar las emociones detectadas dentro del texto analizado. Estos requerimientos son esenciales para garantizar la funcionalidad integral de la herramienta desarrollada, permitiendo automatizar procesos complejos de análisis textual mediante inteligencia artificial y aprendizaje profundo.

Código	Requerimiento funcional	Descripción
RF01	Ingreso de texto	El sistema debe permitir al usuario ingresar textos en idioma español para su análisis
RF02	Preprocesamiento automático	El sistema debe limpiar, normalizar y preparar el texto ingresado antes del análisis
RF03	Tokenización del texto	El sistema debe dividir el texto en unidades léxicas para su procesamiento
RF04	Generación de embeddings	El sistema debe transformar el texto en representaciones vectoriales contextuales.

Código	Requerimiento funcional	Descripción
RF05	Clasificación multietiqueta	El sistema debe identificar uno o varios indicadores de neuroticismo en un mismo texto.
RF06	Visualización de resultados	El sistema debe mostrar al usuario los resultados obtenidos de forma clara.
RF07	Evaluación del modelo	El sistema debe calcular métricas de rendimiento del modelo
RF08	Historial de pruebas	El sistema debe almacenar resultados de pruebas realizadas
RF09	Interfaz interactiva	El sistema debe permitir la interacción mediante una interfaz amigable

Tabla 3. Requerimientos Funcionales. Elaboración propia

Cada uno de estos requerimientos funcionales representa una parte esencial dentro del funcionamiento general del sistema, ya que permiten automatizar tareas complejas relacionadas con el análisis textual y la detección de patrones lingüísticos. Su correcta implementación garantiza que el modelo pueda ejecutarse de forma eficiente, generando resultados confiables y útiles para el usuario final. Además, permiten establecer una interacción adecuada entre el usuario y el sistema, fortaleciendo la aplicabilidad práctica de la solución propuesta.

2.4.2. Requerimientos no Funcionales

Los requerimientos no funcionales describen las características de calidad y restricciones que debe cumplir el sistema para garantizar un funcionamiento

eficiente, seguro y confiable. A diferencia de los requerimientos funcionales, estos no definen acciones específicas del sistema, sino atributos relacionados con el rendimiento, la usabilidad, la seguridad, la disponibilidad y la escalabilidad. En el presente proyecto, estos requerimientos son fundamentales para asegurar que la herramienta desarrollada pueda operar de manera adecuada en diferentes entornos y bajo distintas condiciones de uso. Además, permiten establecer estándares de calidad que contribuyen a mejorar la experiencia del usuario y la efectividad del sistema.

El cumplimiento de los requerimientos no funcionales permite que el sistema no solo realice correctamente sus funciones principales, sino que también ofrezca estabilidad, rapidez y facilidad de uso. Estos atributos resultan importantes en aplicaciones basadas en inteligencia artificial, ya que el procesamiento de datos y la generación de resultados deben realizarse de manera eficiente. Asimismo, garantizan que la solución tecnológica pueda adaptarse a futuras mejoras o ampliaciones, como la incorporación de nuevos rasgos de personalidad o el aumento en el volumen de datos procesados. Por ello, estos requerimientos complementan a los funcionales y fortalecen la calidad general del sistema propuesto.

Código	Requerimiento no funcional	Descripción
RNF01	Rendimiento	El sistema debe procesar y mostrar resultados en un tiempo corto.
RNF02	Disponibilidad	El sistema debe estar disponible durante las pruebas y validaciones.
RNF03	Usabilidad	La interfaz debe ser intuitiva y fácil de usar.
RNF04	Escalabilidad	El sistema debe permitir futuras ampliaciones o mejoras.
RNF05	Compatibilidad	Debe ejecutarse en entornos como Google Colab o navegador web.

Código	Requerimiento no funcional	Descripción
RNF06	Seguridad	Debe proteger la integridad de los datos ingresados.
RNF07	Mantenibilidad	El código debe permitir modificaciones y mantenimiento.
RNF08	Portabilidad	El sistema debe poder ejecutarse en diferentes dispositivos o plataformas.

Tabla 4. Requerimiento No funcionales. Elaboración Propia

El cumplimiento de estos requerimientos permite que el sistema no solo ejecute correctamente sus funciones principales, sino que además ofrezca estabilidad, rapidez y facilidad de uso durante su operación. Estos atributos son especialmente importantes en aplicaciones basadas en inteligencia artificial, donde el procesamiento de grandes volúmenes de datos requiere eficiencia computacional. Asimismo, estos requerimientos permiten que la solución desarrollada pueda adaptarse a futuras mejoras, tales como la incorporación de nuevos rasgos de personalidad o la ampliación del conjunto de datos.

CAPÍTULO 3. COMPONENTES DE LA PROPUESTA

La propuesta desarrollada consiste en un sistema inteligente para la identificación automática de indicadores lingüísticos asociados al rasgo de neuroticismo en textos escritos en idioma español, mediante la integración de técnicas de Procesamiento de Lenguaje Natural (PLN) y modelos de aprendizaje profundo basados en arquitecturas Transformer. El sistema fue diseñado siguiendo una arquitectura modular que organiza cada una de las etapas necesarias para el procesamiento de la información, desde la construcción del conjunto de datos hasta la presentación de los resultados obtenidos al usuario.

El desarrollo de la propuesta comprendió diversas fases que permitieron construir una solución capaz de analizar comentarios en español e identificar automáticamente los indicadores emocionales relacionados con el neuroticismo. Estas fases incluyeron la integración y preparación del conjunto de datos, el proceso de reetiquetado de la información, el preprocesamiento lingüístico, el entrenamiento de diferentes modelos Transformer, el diseño de la interfaz gráfica y la validación del sistema mediante pruebas experimentales.

Cada uno de los componentes fue desarrollado de manera independiente pero interconectada, permitiendo establecer un flujo continuo de procesamiento desde la entrada del comentario hasta la generación de la clasificación final. Esta organización modular facilita el mantenimiento, la reutilización de componentes y la incorporación de futuras mejoras, como la integración de nuevos modelos de lenguaje o la ampliación del conjunto de datos.

En las siguientes subsecciones se describen detalladamente cada uno de los componentes desarrollados durante la investigación, explicando las actividades realizadas, las herramientas utilizadas y la forma en que cada etapa contribuyó al funcionamiento integral del sistema propuesto.

3.1. Construcción del conjunto de datos

El desarrollo del modelo de inteligencia artificial requirió la construcción de un conjunto de datos específico para la identificación automática de indicadores lingüísticos asociados al rasgo de neuroticismo en idioma español. Debido a que no se encontró un dataset público que reuniera las características necesarias para esta investigación, fue necesario realizar un proceso de integración, selección, reestructuración y reetiquetado de información proveniente de diferentes fuentes públicas.

Para ello se utilizaron tres conjuntos de datos obtenidos desde la plataforma Kaggle, los cuales contenían comentarios escritos en idioma español relacionados con emociones, estados mentales y comportamientos expresados por los usuarios. Cada uno de estos conjuntos aportó información complementaria para la construcción del dataset final, permitiendo incrementar la diversidad lingüística y mejorar la representación de los indicadores emocionales analizados durante la investigación.

Una vez recopilada la información, todos los registros fueron integrados en un único conjunto de datos denominado DATA_CURATION, el cual sirvió como base para las etapas posteriores de depuración, análisis y entrenamiento de los modelos de aprendizaje profundo. Durante este proceso se eliminaron registros duplicados, se verificó la consistencia de los datos y se unificó la estructura de las columnas para garantizar la compatibilidad entre las diferentes fuentes de información.

Posteriormente, se realizó un proceso de revisión y organización de las etiquetas originales con el propósito de adaptarlas al objetivo específico de la investigación. Esta etapa permitió transformar un conjunto de datos orientado al análisis general de emociones en un dataset especializado para la identificación automática de indicadores lingüísticos asociados al neuroticismo.

Finalmente, el conjunto de datos consolidado constituyó la base sobre la cual se desarrollaron las etapas de preprocesamiento, entrenamiento y evaluación de los modelos Transformer implementados en el sistema.

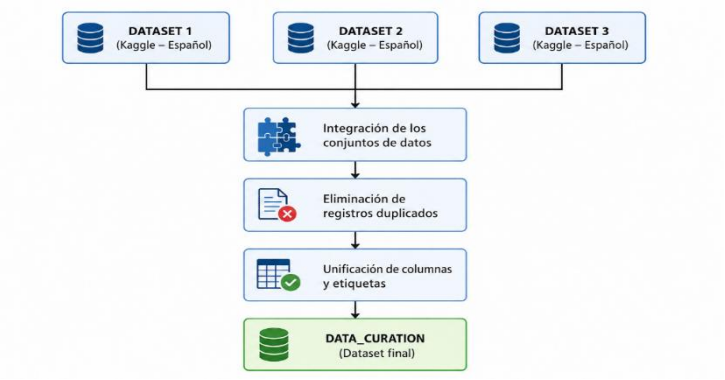


Figura 10. Proceso de integración de los conjuntos de datos empleados para la construcción del dataset final. Elaboración propia.

Definición de los indicadores lingüísticos de neuroticismo

Una vez integrado el conjunto de datos inicial, fue necesario definir las categorías emocionales que serían utilizadas durante el entrenamiento del modelo de clasificación. Debido a que los datasets originales presentaban diferentes estructuras, etiquetas y descripciones psicológicas, se desarrolló un proceso de análisis y reorganización de la información con el propósito de construir un conjunto de etiquetas homogéneo y orientado al objetivo de la investigación.

Para ello se tomó como referencia la relación existente entre diversos rasgos de personalidad, estados emocionales y estados mentales descritos en la literatura psicológica, estableciendo una correspondencia entre los rasgos presentes en los datasets originales y los indicadores lingüísticos asociados al neuroticismo. Este proceso permitió identificar aquellas características psicológicas que representan manifestaciones frecuentes de inestabilidad emocional y que pueden evidenciarse mediante el lenguaje escrito.

Inicialmente, los conjuntos de datos contenían diferentes rasgos psicológicos como lie, antagonism, self_distrust, need_for_acclaim, shame, superiority, cynicism, manipulation, detachment, toughness, aggression, impulsivity y risky_behavior, entre otros. Sin embargo, estas etiquetas no representaban directamente las categorías emocionales que se pretendían identificar en el presente trabajo, por lo que fue necesario agruparlas según su comportamiento psicológico y su relación con el rasgo de neuroticismo.

Como resultado de este proceso se definieron seis indicadores principales: vergüenza, irritabilidad, inseguridad, depresión, ira y ansiedad, los cuales constituyen las etiquetas utilizadas durante el entrenamiento y evaluación de los modelos de inteligencia artificial. Cada uno de estos indicadores representa un conjunto de manifestaciones emocionales que pueden coexistir dentro de un mismo comentario, razón por la cual el problema fue abordado como una tarea de clasificación multietiqueta.

Con el propósito de establecer un criterio uniforme para el proceso de reetiquetado, se realizó un análisis de la relación existente entre los rasgos psicológicos presentes en los datasets originales y los indicadores lingüísticos asociados al neuroticismo. Para ello se tomó como referencia la correspondencia entre la Tríada Oscura de la Personalidad, los rasgos conductuales, las emociones predominantes, los sentimientos asociados y los estados mentales descritos en la literatura especializada.

A partir de este análisis fue posible identificar aquellos rasgos psicológicos que presentaban una relación directa con las categorías emocionales definidas para la presente investigación, permitiendo construir un esquema de clasificación coherente y adaptado al objetivo del estudio.

Personalidad	Rasgo	Trait	Indicador de neuroticismo
Narcisismo	Mentir	Lie	Vergüenza
Narcisismo	Antagonismo	Antagonism	Irritabilidad
Narcisismo	Desconfianza en sí mismo	self_distrust	Depresión
Narcisismo	Necesidad de aclamación	need_for_acclaim	Inseguridad
Narcisismo	Vergüenza	Shame	Vergüenza
Narcisismo	Superioridad	Superiority	Inseguridad

Personalidad	Rasgo	Trait	Indicador de neuroticismo
Maquiavelismo	Cinismo	Cynicism	Irritabilidad
Maquiavelismo	Manipulación	Manipulation	Vergüenza
Maquiavelismo	Desapego	Detachment	Depresión
Psicopatía	Dureza	Toughness	Irritabilidad
Psicopatía	Agresión	Aggression	Ira
Psicopatía	Conducta riesgosa	risky_behavior	Ira
Psicopatía	Impulsividad	Impulsivity	Ansiedad

Tabla 5. Correspondencia entre los rasgos psicológicos originales y los indicadores lingüísticos de neuroticismo. Elaboración Propia

Con base en esta correspondencia se redefinieron las etiquetas utilizadas durante el entrenamiento del modelo, agrupando los diferentes rasgos psicológicos en seis indicadores lingüísticos principales: vergüenza, irritabilidad, inseguridad, depresión, ira y ansiedad. Debido a que un mismo comentario puede expresar simultáneamente varios estados emocionales, el problema fue abordado como una tarea de clasificación multietiqueta.

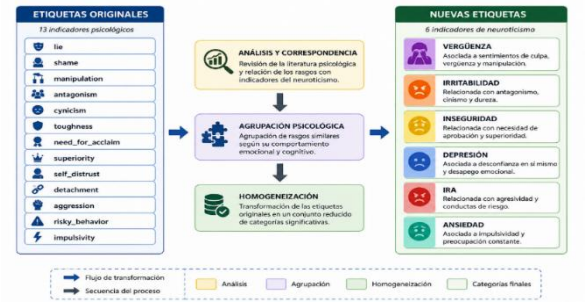


Figura 11. Proceso de transformación de etiquetas.

Análisis y visualización del conjunto de datos

Antes de realizar el preprocesamiento y entrenamiento de los modelos, se efectuó un análisis exploratorio del conjunto de datos con el propósito de conocer su

composición y las principales características de los comentarios utilizados en la investigación. Para ello, se analizó la distribución de los seis indicadores lingüísticos asociados al neuroticismo, la longitud de los textos y la frecuencia de las palabras presentes en cada categoría. Este análisis permitió identificar diferencias entre los indicadores y obtener una visión general de los datos antes de iniciar el proceso de entrenamiento.

La distribución de los indicadores evidenció diferencias en la cantidad de comentarios asociados a cada categoría. Depresión presentó la mayor frecuencia con 4740 registros, seguida de irritabilidad con 4035 y vergüenza con 3968. Por su parte, inseguridad registró 3123 comentarios, ira 2886 y ansiedad presentó la menor frecuencia con 1746 registros. Debido a la naturaleza multietiqueta del conjunto de datos, estas cantidades no representan grupos mutuamente excluyentes, puesto que un mismo comentario puede estar asociado a más de un indicador. La distribución observada permitió identificar el desbalance existente entre las categorías, aspecto considerado posteriormente durante el entrenamiento de los modelos.

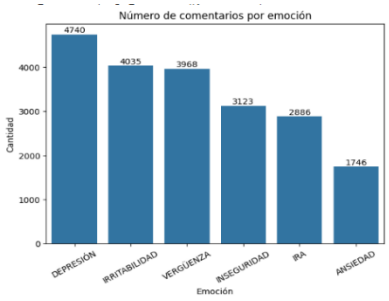


Figura 12. Distribución de los indicadores lingüísticos asociados al neuroticismo en el conjunto de datos. Elaboración propia.

Además de la distribución de las categorías, se analizó la longitud de los comentarios asociados a cada indicador. La representación obtenida muestra una mayor concentración de textos de corta extensión en las seis categorías, aunque se observan diferencias en la dispersión y presencia de comentarios de mayor longitud. Este análisis permitió conocer las características generales de los textos utilizados y considerar su extensión durante las etapas posteriores de preprocesamiento y tokenización requeridas por los modelos Transformer.

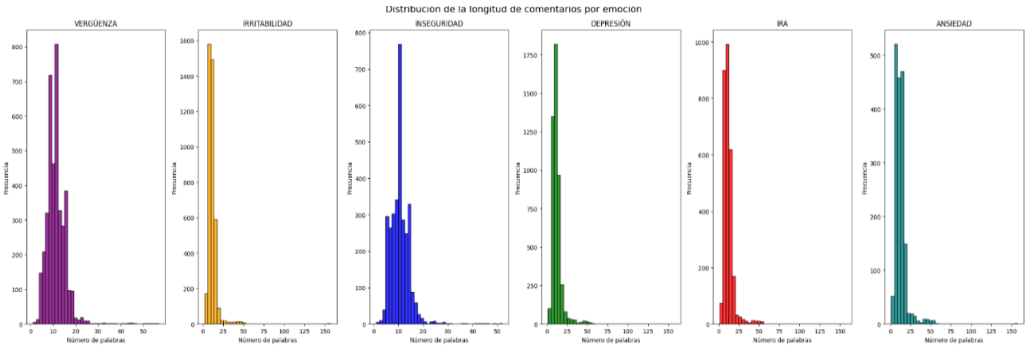


Figura 13. Distribución de la longitud de los comentarios por indicador lingüístico. Elaboración propia.

Finalmente, se realizó una exploración de las palabras presentes en los comentarios correspondientes a cada indicador mediante nubes de palabras. Esta representación permitió visualizar los términos con mayor presencia dentro de los textos asociados a vergüenza, irritabilidad, inseguridad, depresión, ira y ansiedad. Las nubes de palabras se utilizaron como un recurso descriptivo para observar características del contenido textual y no como un criterio de clasificación, debido a que laidentificación de los indicadores depende del contexto completo en el que aparecen las expresiones.

Figura 14. Nubes de palabras correspondientes a los seis indicadores lingüísticos asociados al neuroticismo. Elaboración propia.

El análisis exploratorio permitió identificar características relevantes del conjunto de datos antes del entrenamiento de los modelos. La diferencia en la cantidad de registros asociados a cada indicador evidenció la presencia de desbalance entre las categorías, mientras que el análisis de la longitud y de las palabras frecuentes permitió conocer de manera general la composición de los comentarios utilizados. Estos resultados sirvieron como referencia para las etapas posteriores de preprocesamiento y configuración del proceso de entrenamiento.

Construcción y balanceo del conjunto de datos

Una vez definidos los indicadores lingüísticos asociados al rasgo de neuroticismo, se procedió a la construcción del conjunto de datos definitivo que sería utilizado durante el entrenamiento y evaluación de los modelos de aprendizaje profundo. Debido a que los datasets originales presentaban diferencias en la cantidad de registros por categoría, fue necesario realizar un proceso de integración, selección, reetiquetado y balanceo de la información con el propósito de obtener un conjunto de datos más uniforme y representativo.

Inicialmente, los comentarios provenientes de los diferentes datasets fueron integrados en un único archivo denominado DATA_CURATION, el cual concentró toda la información recopilada durante la fase de preparación de datos. Posteriormente, cada registro fue revisado y asociado con uno o varios indicadores lingüísticos de neuroticismo, considerando la relación establecida entre los rasgos psicológicos originales y las categorías definidas para la investigación.

Con el objetivo de reducir el desbalance existente entre las diferentes clases, se realizó un proceso de selección controlada de registros provenientes de los datasets complementarios. Esta actividad fue desarrollada siguiendo los criterios establecidos durante la dirección de la investigación, incorporando nuevos comentarios para aquellas categorías que presentaban menor cantidad de ejemplos. De esta manera, se buscó mantener una distribución más equilibrada entre las etiquetas utilizadas durante el entrenamiento del modelo.

Para completar la distribución del conjunto de datos se incorporaron registros correspondientes a las categorías de depresión, ansiedad, ira e inseguridad, seleccionados aleatoriamente a partir de los datasets complementarios. Este procedimiento permitió incrementar la representatividad de dichas categorías y mejorar la distribución general del dataset, reduciendo el sesgo producido por clases con poca frecuencia de aparición.

Como resultado de este proceso se obtuvo un conjunto de datos multietiqueta conformado por comentarios escritos en idioma español, donde cada registro puede pertenecer simultáneamente a uno o varios indicadores de neuroticismo. Esta característica refleja de forma más adecuada la naturaleza del lenguaje humano, ya

que un mismo comentario puede expresar diferentes estados emocionales de manera simultánea.

El proceso de integración y balanceo permitió obtener un conjunto de datos especializado para la identificación automática de indicadores lingüísticos asociados al rasgo de neuroticismo. Debido a la naturaleza multietiqueta del problema, un mismo comentario puede estar relacionado simultáneamente con uno o varios indicadores emocionales, lo que genera una distribución diferente en la cantidad de registros por cada categoría.

La construcción del dataset definitivo permitió disponer de información suficiente para el entrenamiento, validación y evaluación de los modelos de aprendizaje profundo implementados durante la investigación. La distribución final de los registros utilizados en cada indicador lingüístico fue presentada previamente en el Capítulo 1, correspondiente al análisis de recolección de datos.

Reetiquetado del conjunto de datos

El conjunto de datos integrado requirió un proceso de reetiquetado con el propósito de adaptar las etiquetas originales al objetivo específico de la presente investigación. Aunque los datasets utilizados contenían información relacionada con emociones, rasgos psicológicos y estados mentales, sus categorías originales no coincidían con los indicadores lingüísticos de neuroticismo definidos para el desarrollo del modelo. Por esta razón, fue necesario establecer un nuevo esquema de clasificación que permitiera representar de forma adecuada las categorías emocionales utilizadas durante el entrenamiento de los modelos de inteligencia artificial.

El proceso de reetiquetado se desarrolló a partir de la correspondencia establecida entre los rasgos psicológicos originales y los indicadores lingüísticos definidos para esta investigación. Con base en esta relación, cada comentario fue revisado y asociado con una o varias categorías emocionales según el contenido expresado en el texto y las características psicológicas previamente identificadas. Este procedimiento permitió transformar las etiquetas originales en un conjunto de

categorías homogéneas orientadas específicamente a la identificación del rasgo de neuroticismo.

Como resultado de este proceso se establecieron seis indicadores lingüísticos principales: vergüenza, irritabilidad, inseguridad, depresión, ira y ansiedad, los cuales constituyen las etiquetas utilizadas durante el entrenamiento y evaluación de los modelos Transformer implementados en esta investigación. Debido a la naturaleza del lenguaje humano, un mismo comentario puede expresar simultáneamente diferentes estados emocionales, razón por la cual el problema fue abordado mediante un esquema de clasificación multietiqueta.

Posteriormente, se generaron las columnas binarias correspondientes a cada uno de los indicadores definidos. Para ello, se asignó el valor de 1 cuando un comentario presentaba evidencia de un determinado indicador lingüístico y el valor de 0 cuando dicho indicador no estaba presente. Esta representación permitió estructurar el conjunto de datos en un formato adecuado para el entrenamiento de modelos de clasificación multietiqueta basados en arquitecturas Transformer.

El proceso de reetiquetado permitió unificar los criterios de clasificación entre los diferentes conjuntos de datos utilizados, obteniendo un dataset especializado para la identificación automática de indicadores lingüísticos asociados al rasgo de neuroticismo en textos escritos en idioma español. Asimismo, esta etapa constituyó la base para las actividades posteriores de preprocesamiento, representación del lenguaje y entrenamiento de los modelos de aprendizaje profundo.

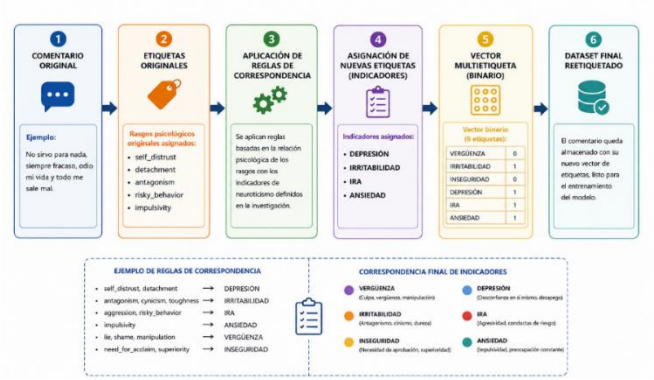


Figura 15. Proceso de reetiquetado del conjunto de datos para la construcción del dataset multietiqueta.

3.2. Preprocesamiento del conjunto de datos

Una vez construido y reetiquetado el conjunto de datos, se desarrolló una etapa de preprocesamiento con el propósito de mejorar la calidad de la información antes del entrenamiento de los modelos de aprendizaje profundo. Esta fase constituye una de las etapas más importantes dentro del procesamiento de lenguaje natural, debido a que permite reducir el ruido presente en los textos, homogenizar la información y preparar los comentarios para su representación mediante modelos Transformer.

El proceso de preprocesamiento fue implementado utilizando el entorno de desarrollo Google Colab, empleando bibliotecas especializadas para el procesamiento de lenguaje natural en idioma español, entre ellas Pandas, NumPy, NLTK, spaCy, Stanza, Clean-Text y expresiones regulares (Regex). Estas herramientas permitieron automatizar las diferentes tareas de limpieza y transformación aplicadas al conjunto de datos.

Inicialmente, se realizó una revisión general del dataset con el fin de identificar registros vacíos, comentarios duplicados y posibles inconsistencias en las etiquetas. Posteriormente, se normalizó la estructura del conjunto de datos, asegurando que todas las variables conservaran un formato homogéneo para facilitar las etapas posteriores del procesamiento.

Durante esta fase también se desarrolló una secuencia de transformaciones orientadas a conservar únicamente la información lingüística relevante para el entrenamiento de los modelos de clasificación. Entre estas actividades se incluyeron la normalización del texto, eliminación de caracteres innecesarios, tratamiento de emojis, limpieza de enlaces, eliminación de espacios adicionales, tokenización, eliminación de palabras vacías y lematización, obteniendo finalmente una representación textual más consistente para el aprendizaje automático.

Cada una de estas actividades fue implementada de forma secuencial, permitiendo que la salida generada en una etapa constituyera la entrada del siguiente proceso de transformación. De esta manera, se obtuvo un conjunto de datos limpio,

normalizado y estructurado, adecuado para el entrenamiento de los modelos BERT, RoBERTa, DistilBERT y Flan-T5 utilizados en la presente investigación.

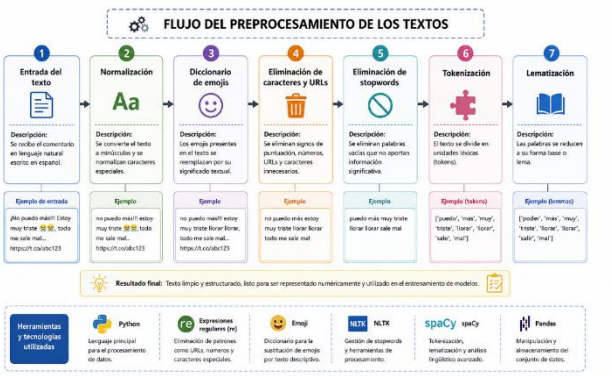


Figura 16. Flujo general del proceso de preprocesamiento aplicado al conjunto de datos. Elaboración propia.

Normalización del texto

La primera etapa del preprocesamiento consistió en la normalización de los comentarios con el objetivo de homogenizar su estructura antes de aplicar las técnicas de Procesamiento del Lenguaje Natural. Debido a que los textos provenían de diferentes fuentes de información, presentaban diferencias en el uso de mayúsculas y minúsculas, caracteres especiales, repeticiones de letras, enlaces web y otros elementos que podían introducir ruido y afectar el desempeño de los modelos de clasificación.

Durante esta etapa, todos los comentarios fueron convertidos a letras minúsculas con el propósito de reducir la variabilidad léxica ocasionada por distintas formas de escritura de una misma palabra. Posteriormente, se eliminaron enlaces URL, caracteres especiales, espacios en blanco innecesarios y secuencias repetitivas que no aportaban información relevante para la identificación de los indicadores lingüísticos de neuroticismo. Asimismo, se conservaron los caracteres propios del idioma español, como las vocales acentuadas y la letra "ñ", preservando el significado original de los textos.

La Figura 17 presenta de manera general las actividades realizadas durante la etapa de normalización. Este procedimiento permitió obtener un conjunto de comentarios

con un formato uniforme, reduciendo el ruido presente en los datos y facilitando la aplicación de las etapas posteriores de limpieza, eliminación de palabras vacías, tokenización y lematización.

Figura 17. Proceso de normalización aplicado a los comentarios del conjunto de datos. Elaboración propia.

Finalmente, la normalización constituyó la base para el resto del preprocesamiento, ya que permitió estandarizar los textos antes de la extracción de características lingüísticas y del entrenamiento de los modelos BERT, RoBERTa, DistilBERT y FLAN-T5, garantizando una representación más consistente de la información utilizada durante la clasificación.

Construcción del diccionario de emojis

Durante el análisis exploratorio del conjunto de datos se observó que una gran cantidad de comentarios contenían emojis utilizados por los usuarios para expresar emociones, estados de ánimo o reacciones frente a diferentes situaciones. Debido a que estos elementos representan información emocional relevante, se decidió incorporarlos al proceso de análisis en lugar de eliminarlos completamente.

Para ello se desarrolló un diccionario de equivalencias que permitió reemplazar cada emoji por una palabra o expresión textual con significado emocional similar. Esta estrategia permitió conservar la información semántica contenida en los emojis y facilitar su interpretación por parte de los modelos de procesamiento de lenguaje natural.

El diccionario fue construido manualmente considerando el contexto emocional representado por cada emoji y su posible relación con los indicadores lingüísticos de neuroticismo definidos para la investigación. Posteriormente, durante el proceso de limpieza del texto, cada emoji identificado fue sustituido automáticamente por su equivalente textual antes de continuar con las siguientes etapas del preprocesamiento.

La incorporación de este diccionario permitió preservar la información emocional expresada mediante emojis, evitando que dichos elementos fueran descartados durante la limpieza del texto. Como resultado, los comentarios conservaron una representación semántica más completa, favoreciendo las etapas posteriores de tokenización, lematización y entrenamiento de los modelos Transformer.

Figura 18. Diccionario de equivalencias utilizado para la sustitución de emojis por categorías emocionales durante el proceso de normalización. Elaboración propia.

Eliminación de caracteres especiales y enlaces

Con el propósito de mejorar la calidad del conjunto de datos, se implementó una etapa de limpieza textual orientada a eliminar aquellos elementos que no aportaban información relevante para la tarea de clasificación. Esta fase permitió reducir el ruido presente en los comentarios y obtener una representación más consistente del lenguaje que posteriormente sería procesado por los modelos de aprendizaje profundo. La depuración de estos elementos facilitó la estandarización de los textos, evitando que información irrelevante influyera durante el entrenamiento de los modelos y mejorando la calidad general de los datos empleados en la investigación.

Durante esta etapa se eliminaron enlaces web (URL), caracteres especiales, signos de puntuación innecesarios, números, espacios en blanco adicionales y otros símbolos sin valor semántico para la identificación de los indicadores lingüísticos asociados al rasgo de neuroticismo. Asimismo, se normalizaron las repeticiones excesivas de caracteres, frecuentes en publicaciones realizadas en redes sociales, con el propósito de conservar únicamente la forma correcta de cada palabra y reducir la variabilidad producida por diferentes estilos de escritura empleados por los usuarios.

La limpieza textual fue implementada mediante funciones desarrolladas en Python utilizando expresiones regulares (Regular Expressions o Regex), lo que permitió automatizar el procesamiento de todos los comentarios del conjunto de datos de forma eficiente y consistente. De manera complementaria, se preservaron los caracteres propios del idioma español, como las vocales acentuadas y la letra «ñ», evitando alterar el significado original de los textos y garantizando que la información lingüística relevante permaneciera disponible para las siguientes etapas del procesamiento.

La Figura 19 muestra de manera general el flujo de actividades ejecutadas durante esta etapa de limpieza textual, desde la eliminación de enlaces y caracteres especiales hasta la obtención de un comentario limpio y normalizado. Este procedimiento permitió reducir la presencia de ruido en los datos y generar una representación textual uniforme, facilitando la aplicación de las etapas posteriores de eliminación de palabras vacías, tokenización y lematización.

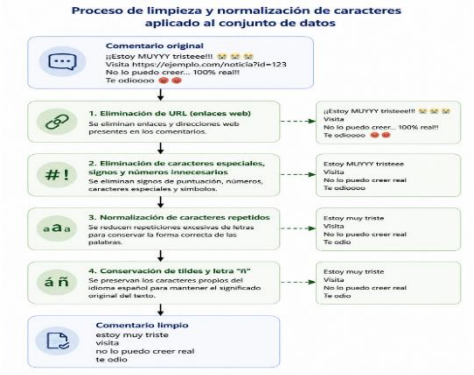


Figura 19. Proceso de limpieza y normalización de caracteres aplicado al conjunto de datos. Elaboración propia.

Finalmente, esta etapa contribuyó a disminuir la variabilidad del texto y a mejorar la calidad de la información utilizada durante el entrenamiento de los modelos BERT, RoBERTa, DistilBERT y FLAN-T5. Como resultado, el conjunto de datos presentó una estructura más consistente y adecuada para la extracción de características lingüísticas, favoreciendo el desempeño de los modelos en la tarea de clasificación multietiqueta de los indicadores de neuroticismo.

Eliminación de palabras vacías (Stopwords)

Una vez finalizada la etapa de normalización y limpieza textual, se procedió a la eliminación de palabras vacías (*stopwords*), cuyo objetivo fue reducir la presencia de términos con escasa carga semántica que no aportaban información relevante para la tarea de clasificación. Este procedimiento permitió disminuir el ruido presente en los comentarios y conservar únicamente aquellas palabras con mayor capacidad para representar el contenido emocional y lingüístico asociado a los indicadores de neuroticismo.

Como base para este proceso se empleó el listado de palabras vacías proporcionado por la biblioteca NLTK para el idioma español. Sin embargo, debido a las características particulares del conjunto de datos, dicho listado fue complementado mediante la construcción de un diccionario personalizado, incorporando expresiones identificadas durante el análisis exploratorio que resultaban frecuentes pero poco informativas para el proceso de clasificación. Esta adaptación permitió ajustar la limpieza textual a las necesidades específicas de la investigación.

De manera complementaria, algunas palabras fueron excluidas deliberadamente del listado de *stopwords* debido a su importancia dentro del contexto emocional analizado. Verbos, pronombres y expresiones relacionadas con estados psicológicos, como *soy*, *estoy*, *siento*, *ansiedad*, *triste* e *inseguro*, fueron conservados por su alto valor semántico para la identificación automática de los indicadores lingüísticos de neuroticismo. La Figura 20 presenta los recursos léxicos utilizados durante esta etapa del preprocesamiento.

Figura 20. Palabras clave conservadas y diccionario personalizado de stopwords utilizados durante el proceso de eliminación de palabras vacías. Elaboración propia.

La utilización de un diccionario personalizado permitió adaptar el proceso de limpieza a las características particulares del conjunto de datos, evitando la eliminación de términos con información relevante para la clasificación multietiqueta. Como resultado, los comentarios conservaron una representación lingüística más significativa, favoreciendo las etapas posteriores de tokenización, lematización y entrenamiento de los modelos Transformer.

Tokenización y lematización del texto

Una vez concluida la limpieza y eliminación de palabras vacías, se procedió a la tokenización y lematización de los comentarios con el propósito de obtener una representación lingüística más adecuada para el entrenamiento de los modelos de aprendizaje profundo. Estas técnicas permitieron transformar los textos en unidades estructuradas, reduciendo la variabilidad del lenguaje y facilitando la extracción de información relevante para la clasificación automática de los indicadores de neuroticismo.

La tokenización consistió en dividir cada comentario en unidades léxicas denominadas tokens, permitiendo analizar individualmente cada palabra que conforma el texto. Para esta tarea se utilizó la biblioteca spaCy, empleando el

modelo lingüístico es_core_news_md, especializado en el procesamiento del idioma español. Esta herramienta permitió realizar la segmentación del texto de manera eficiente, conservando la estructura gramatical de cada comentario.

Posteriormente, se aplicó el proceso de lematización, mediante el cual cada palabra fue transformada a su forma canónica o lema, independientemente de su género, número o conjugación verbal. De forma complementaria, se realizó un análisis morfosintáctico utilizando las etiquetas gramaticales (Part of Speech o POS) generadas por spaCy, eliminando automáticamente determinadas categorías gramaticales, como preposiciones, conjunciones y signos de puntuación que no aportaban información significativa para la clasificación. La Figura 21 muestra de manera general el flujo de este procedimiento.

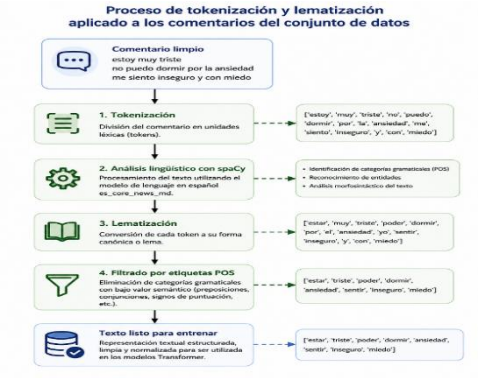


Figura 21. Proceso de tokenización y lematización aplicado a los comentarios del conjunto de datos. Elaboración propia.

Como resultado de esta etapa se obtuvo una representación textual más limpia, estructurada y consistente, optimizando la calidad de la información utilizada durante el entrenamiento de los modelos BERT, RoBERTa, DistilBERT y FLAN-T5. Esta transformación permitió reducir la variabilidad lingüística de los comentarios y favorecer una representación semántica más uniforme para la tarea de clasificación multietiqueta.

Generación del conjunto de datos procesados

Como resultado de todas las etapas de preprocesamiento, se generó un nuevo conjunto de datos con diferentes representaciones de cada comentario, permitiendo

conservar la información obtenida durante las fases de limpieza, normalización, eliminación de palabras vacías, tokenización y lematización. Estas transformaciones facilitaron el análisis exploratorio de los datos y garantizaron una preparación adecuada de la información antes del entrenamiento de los modelos de aprendizaje profundo.

Durante este procedimiento se incorporaron nuevas columnas al conjunto de datos original, correspondientes al texto limpio (COMENTARIOS_LIMPIOS), los comentarios sin palabras vacías (COMENTARIOS_SIN_STOPWORDS) y la representación tokenizada (COMENTARIOS_TOKENIZADOS). La organización de estas variables permitió mantener un registro completo de cada transformación aplicada sobre los comentarios, facilitando la validación de las distintas etapas del preprocesamiento.

La Figura 22 presenta la estructura del conjunto de datos una vez finalizado el proceso de preparación de la información. En ella se observa la incorporación de las nuevas representaciones textuales generadas automáticamente, las cuales constituyen la base para la construcción de las entradas utilizadas durante el entrenamiento de los modelos BERT, RoBERTa, DistilBERT y FLAN-T5.

	COMENTARIOS	VERGÜENZA	IRRITABILIDAD	INSEGURIDAD	DEPRESIÓN	IRA	ANSIEDAD	COMENTARIOS_LIMPIOS	COMENTARIOS_SIN_STOPWORDS	COMENTARIOS_TOKENIZADOS
0	No puedo creer la vergüenza que siento por hab...	1	0	1	0	0	0	no puedo creer la vergüenza que siento por hab...	[no, puedo, creer, vergüenza, que, siento, hab...	[no, puedo, creer, la, vergüenza, que, siento,...
1	Lamentablemente he pasado vergüenza al hablar ...	1	1	0	0	1	1	lamentablemente he pasado vergüenza al hablar ...	[lamentablemente, pasado, vergüenza, hablar, m...	[lamentablemente, he, pasado, vergüenza, al, h...
2	Es humillante mentir sobre mi ubicación de for...	1	1	0	0	0	0	es humillante mentir sobre mi ubicación de for...	[es, humillante, mentir, mi, ubicación, forma,...	[es, humillante, mentir, sobre, mi, ubicación,...
3	Me da vergüenza haber impedir tus planes hace ...	1	0	0	0	1	1	me da vergüenza haber impedir tus planes hace ...	[da, vergüenza, haber, impedir, planes, hace, ...	[me, da, vergüenza, haber, impedir, tus, plane...
4	Qué vergüenza descuidar tu petición en la reun...	1	0	0	0	1	0	qué vergüenza descuidar tu petición en la reun...	[vergüenza, descuidar, petición, reunión, lame...	[qué, vergüenza, descuidar, tu, petición, en, ...
...	...	...	...	...	...	...	...	...	...	...
1923	Estoy solo, incluso cuando estoy rodeado.	1	1	0	1	0	0	estoy solo incluso cuando estoy rodeado	[estoy, solo, incluso, estoy, rodeado]	[estoy, solo, incluso, cuando, estoy, rodeado]
1924	Me volví espectador de mi propia vida.	0	1	0	1	0	0	me volví espectador de mi propia vida	[volví, espectador, mi, propia, vida]	[me, volví, espectador, de, mi, propia, vida]
1925	El dolor es la única constante que conozco.	0	1	0	1	0	0	el dolor es la única constante que conozco	[dolor, es, única, constante, que, conozco]	[el, dolor, es, la, única, constante, que, con...
1926	Estoy harto de fingir que tengo el control.	1	1	0	1	0	0	estoy harto de fingir que tengo el control	[estoy, harto, fingir, tengo, control]	[estoy, harto, de, fingir, que, tengo, el, con...
1927	Cada día es una repetición del mismo vacío.	0	1	0	1	0	0	cada día es una repetición del mismo vacío	[cada, día, es, repetición, mismo, vacío]	[cada, día, es, una, repetición, del, mismo, v...

1926 rows × 10 columns

Figura 22. Estructura del conjunto de datos después del proceso de preprocesamiento. Elaboración propia.

Finalmente, el conjunto de datos procesado constituyó la versión definitiva utilizada para la división en los conjuntos de entrenamiento y prueba. La incorporación de estas nuevas representaciones permitió disponer de información más estructurada y consistente, favoreciendo la extracción de características lingüísticas y

contribuyendo a mejorar la calidad de los datos empleados durante la etapa de aprendizaje de los modelos Transformer.

Selección de los modelos Transformer

Una vez concluido el proceso de preprocesamiento del conjunto de datos, se procedió a seleccionar los modelos de aprendizaje profundo utilizados para la identificación automática de indicadores lingüísticos asociados al rasgo de neuroticismo en comentarios escritos en idioma español. La selección consideró la capacidad de las arquitecturas Transformer para representar el contexto semántico del lenguaje, su desempeño reportado en tareas de clasificación textual y la disponibilidad de modelos preentrenados en español a través de la biblioteca Hugging Face Transformers. Estos criterios permitieron elegir arquitecturas ampliamente utilizadas en investigaciones recientes relacionadas con el procesamiento automático del lenguaje natural.

Con el propósito de analizar diferentes enfoques de procesamiento, se seleccionaron cuatro modelos: BERT, RoBERTa, DistilBERT y Flan-T5. Cada uno presenta características particulares en cuanto a arquitectura, estrategia de adaptación y requerimientos computacionales, permitiendo evaluar distintas alternativas para abordar el problema de clasificación multietiqueta planteado en esta investigación. Esta diversidad facilitó comparar modelos especializados en tareas de clasificación con un modelo generativo basado en instrucciones, proporcionando una visión más amplia sobre su comportamiento en textos escritos en español.

Los modelos BERT, RoBERTa y DistilBERT corresponden a arquitecturas basadas principalmente en codificadores (*Encoder*), ampliamente utilizadas en tareas de clasificación textual debido a su capacidad para generar representaciones contextuales del lenguaje. En esta investigación fueron adaptados mediante la técnica de *Fine-Tuning*, utilizando el conjunto de datos previamente construido para identificar los seis indicadores lingüísticos asociados al neuroticismo: vergüenza, irritabilidad, inseguridad, depresión, ira y ansiedad. Este procedimiento permitió

ajustar los parámetros de cada modelo a las características particulares del problema estudiado.

Por su parte, Flan-T5 corresponde a una arquitectura *Encoder–Decoder* desarrollada mediante *Instruction Tuning*, diseñada para interpretar instrucciones expresadas en lenguaje natural y generar respuestas a partir de ejemplos proporcionados en el *prompt*. Debido a estas características, el modelo fue incorporado como una estrategia de Few-Shot Learning, sin realizar un proceso de ajuste de parámetros (*Fine-Tuning*). Su incorporación permitió evaluar el potencial de los modelos generativos en escenarios con disponibilidad limitada de ejemplos de entrenamiento y comparar su comportamiento frente a los modelos tradicionales basados en codificadores.

La incorporación de estas cuatro arquitecturas permitió analizar diferentes estrategias de aprendizaje profundo aplicadas a la detección automática de indicadores lingüísticos de neuroticismo. Mientras que BERT, RoBERTa y DistilBERT fueron entrenados mediante *Fine-Tuning* utilizando el conjunto completo de entrenamiento, Flan-T5 fue evaluado mediante inferencia *Few-Shot* sobre un subconjunto representativo de comentarios. En consecuencia, los resultados obtenidos permitieron identificar las fortalezas y limitaciones de cada enfoque, considerando las diferencias metodológicas existentes entre ambas estrategias de adaptación.

Con el propósito de sintetizar las principales características de cada arquitectura, la **¡Error! No se encuentra el origen de la referencia.** presenta un resumen de los modelos Transformer implementados en esta investigación, indicando el tipo de arquitectura, la estrategia de adaptación empleada y la principal característica que motivó su selección. De manera complementaria, la Figura 23Figura 23 ilustra gráficamente las arquitecturas y estrategias de adaptación utilizadas, facilitando la comprensión de las diferencias entre los modelos basados en codificadores y el enfoque generativo representado por Flan-T5.

Modelo	Arquitectura	Tipo de entrenamiento	Característica principal
BERT	Encoder	Fine-Tuning	Comprensión bidireccional del contexto.
RoBERTa	Encoder	Fine-Tuning	Optimización del preentrenamiento de BERT
DistilBERT	Encoder	Fine-Tuning	Menor costo computacional y alta eficiencia.
Flan-T5	Encoder–Decoder	Instruction Tuning + Few-Shot	Modelo generativo capaz de seguir instrucciones y adaptarse a nuevas tareas con pocos ejemplos.

Tabla 6. Características de los modelos Transformer implementados en la investigación. Elaboración propia.

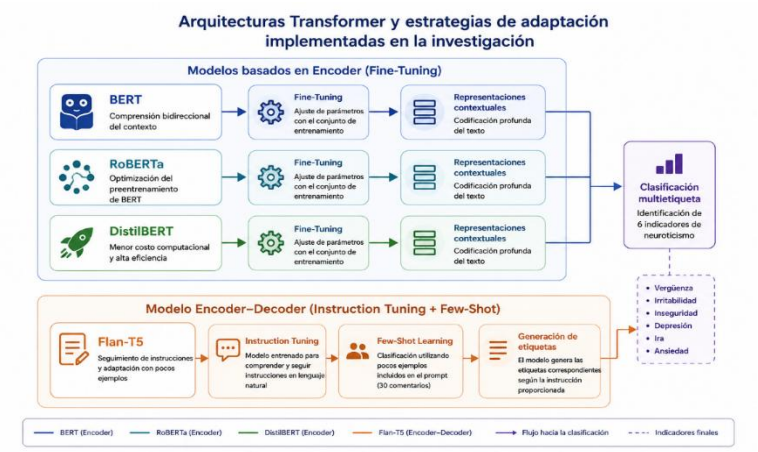


Figura 23. Arquitecturas Transformer y estrategias de adaptación implementadas en la investigación. Elaboración propia

3.3. Entrenamiento de los modelos

Una vez finalizado el proceso de preprocesamiento y seleccionadas las arquitecturas Transformer, se inició la etapa de entrenamiento de los modelos de aprendizaje profundo. Esta fase tuvo como propósito adaptar los modelos preentrenados al problema específico de la detección automática de indicadores lingüísticos asociados al rasgo de neuroticismo en comentarios escritos en español. Para ello, se empleó el conjunto de datos procesado, garantizando que toda la

información utilizada durante el entrenamiento presentara una estructura homogénea y adecuada para las tareas de clasificación multietiqueta.

El entrenamiento fue desarrollado utilizando el entorno de ejecución de Google Colaboratory, aprovechando el procesamiento acelerado mediante unidades gráficas (GPU) para reducir el tiempo requerido durante el ajuste de los modelos. Asimismo, se emplearon bibliotecas especializadas como PyTorch, Transformers de Hugging Face y Scikit-learn, las cuales proporcionaron las herramientas necesarias para la carga de los modelos preentrenados, la optimización de los parámetros y la evaluación del desempeño durante cada época de entrenamiento.

Con el propósito de garantizar una comparación consistente entre los modelos basados en *Fine-Tuning*, se mantuvieron configuraciones de entrenamiento similares para BERT, RoBERTa y DistilBERT, incluyendo el número de épocas, el optimizador, la función de pérdida y la estrategia de partición del conjunto de datos. En el caso de Flan-T5, debido a su naturaleza generativa, la evaluación se realizó mediante la estrategia *Few-Shot Learning*, empleando instrucciones y ejemplos incluidos en el *prompt*, sin modificar los parámetros internos del modelo.

La Figura 24 presenta de manera general el flujo seguido durante la etapa de entrenamiento de los modelos implementados en la investigación. Este procedimiento permitió mantener una metodología uniforme para las arquitecturas ajustadas mediante *Fine-Tuning* y, al mismo tiempo, incorporar un enfoque alternativo basado en aprendizaje con pocos ejemplos para evaluar el comportamiento de Flan-T5.

Figura 24. Flujo general del proceso de entrenamiento de los modelos Transformer implementados en la investigación. Elaboración propia.

Finalmente, los modelos entrenados fueron evaluados utilizando diferentes métricas de clasificación, entre ellas precisión, *recall*, F1-score, exactitud y Hamming Loss. Los resultados obtenidos permitieron analizar el desempeño individual de cada arquitectura y determinar cuál ofreció el mejor comportamiento para la detección automática de indicadores lingüísticos de neuroticismo en textos escritos en español.

Resumen de hiperparámetros

El entrenamiento de los modelos constituyó una de las etapas principales de la propuesta, debido a que permitió adaptar las arquitecturas Transformer seleccionadas a la tarea específica de detección automática de indicadores lingüísticos asociados al neuroticismo en textos escritos en español. Para esta etapa se utilizaron los modelos BERT, RoBERTa y DistilBERT, los cuales fueron ajustados mediante *Fine-Tuning* utilizando el conjunto de datos previamente procesado.

Con el propósito de realizar una evaluación bajo condiciones experimentales comparables, los tres modelos fueron entrenados utilizando la misma configuración general de hiperparámetros. Se establecieron cinco épocas de entrenamiento, una tasa de aprendizaje de, un tamaño de lote de 8 y un *weight decay* de 0.01. Asimismo, se utilizó el optimizador AdamW y la función de pérdida BCEWithLogitsLoss con ponderación de clases, considerando la naturaleza multietiqueta del problema y el desbalance existente entre los indicadores lingüísticos. La Tabla 7 presenta un resumen de la configuración utilizada.

Hiperparámetro	BERT	RoBERTa	DistilBERT
Tasa de aprendizaje (*Learning rate*)	2×10^{-5}	2×10^{-5}	2×10^{-5}
Tamaño de lote (*Batch size*)	8	8	8

Hiperparámetro	BERT	RoBERTa	DistilBERT
Número de épocas	5	5	5
Weight decay	0.01	0.01	0.01
Optimizador	AdamW	AdamW	AdamW
Función de pérdida	BCEWithLogitsLoss ponderada	BCEWithLogitsLoss ponderada	BCEWithLogitsLoss ponderada
Número de etiquetas	6	6	6

Tabla 7. Resumen de hiperparámetros utilizados durante el entrenamiento de los modelos mediante Fine-Tuning. Elaboración propia.

La utilización de una configuración común permitió analizar el comportamiento de las tres arquitecturas bajo condiciones equivalentes y reducir la influencia de variaciones en los parámetros de entrenamiento durante la comparación de los resultados. Las diferencias obtenidas en las métricas de evaluación se analizaron posteriormente considerando las características propias de cada arquitectura.

División del conjunto de datos

Con el propósito de evaluar el desempeño de los modelos de aprendizaje profundo sobre información no utilizada durante el entrenamiento, el conjunto de datos fue dividido en dos subconjuntos independientes: entrenamiento y prueba. Esta estrategia permitió ajustar los parámetros de los modelos utilizando una parte de la información disponible y, posteriormente, evaluar su capacidad de generalización sobre comentarios previamente no observados. La separación de los datos constituye una práctica ampliamente utilizada en tareas de aprendizaje supervisado, ya que reduce el riesgo de sobreajuste y permite obtener una evaluación más objetiva del rendimiento de los modelos.

Para realizar esta partición se utilizó la función train_test_split de la biblioteca Scikit-learn, estableciendo una distribución del 80 % para entrenamiento y 20 % para prueba. Asimismo, se empleó un valor fijo de random_state = 42, garantizando

que la división pudiera reproducirse en futuras ejecuciones y permitiendo obtener resultados consistentes durante el proceso experimental. Como variables de entrada se utilizaron los comentarios previamente preprocesados (*COMENTARIOS_LIMPIOS*), mientras que las variables de salida correspondieron a las seis etiquetas emocionales definidas para la clasificación multietiqueta.

La Figura 25 presenta la implementación realizada para la división del conjunto de datos, donde puede observarse la configuración utilizada y el número de registros obtenidos para cada subconjunto. Como resultado del procedimiento, el conjunto de entrenamiento quedó conformado por 6489 comentarios, mientras que el conjunto de prueba estuvo integrado por 1623 comentarios, manteniendo la proporción establecida para el desarrollo de la investigación.

```
from sklearn.model_selection import train_test_split

# X = textos limpios
X = dataset_read["COMENTARIOS_LIMPIOS"]

# Y = columnas multilabel
y = dataset_read[["VERGÜENZA", "IRRITABILIDAD", "INSEGURIDAD",
                  "DEPRESIÓN", "IRA", "ANSIEDAD"]]

# División 80% train - 20% test
X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.2, random_state=42
)

print("Train size:", len(X_train))
print("Test size:", len(X_test))
```

```
Train size: 6489
Test size: 1623
```

Figura 25. Implementación de la división del conjunto de datos en entrenamiento (80 %) y prueba (20 %) utilizando la función train_test_split. Elaboración propia.

La utilización de esta estrategia permitió entrenar los modelos Transformer sobre una muestra representativa del conjunto de datos y, posteriormente, evaluar su capacidad de clasificación utilizando información completamente independiente. De esta manera, los resultados obtenidos reflejan con mayor precisión el comportamiento esperado de los modelos frente a nuevos comentarios, fortaleciendo la validez de la evaluación experimental desarrollada en la presente investigación.

Ajuste por pesos de clase (Weighted Loss)

Antes del entrenamiento de los modelos Transformer se realizó un análisis de la distribución de las etiquetas presentes en el conjunto de datos con el propósito de

identificar posibles diferencias en la frecuencia de aparición de cada indicador lingüístico. Este análisis permitió comprobar que las seis categorías emocionales no presentaban una distribución uniforme, evidenciando un desbalance entre clases que podía afectar el proceso de aprendizaje y favorecer la predicción de las categorías con mayor número de ejemplos. Debido a ello, fue necesario incorporar una estrategia que compensara estas diferencias durante el entrenamiento y contribuyera a un aprendizaje más equilibrado.

Los resultados obtenidos mostraron que las etiquetas depresión, irritabilidad y vergüenza concentraban un mayor número de registros, mientras que categorías como ansiedad e ira presentaban una frecuencia considerablemente menor. Con base en esta distribución se calcularon pesos específicos para cada categoría utilizando la función compute_class_weight de la biblioteca Scikit-learn, configurada con el criterio balanced, permitiendo asignar automáticamente una mayor importancia a las clases menos representadas y reduciendo el efecto del desbalance existente en el conjunto de datos.

La Figura 26 presenta el procedimiento implementado para obtener la distribución de las etiquetas emocionales y calcular los pesos utilizados durante el entrenamiento de los modelos. Este proceso permitió generar un conjunto de pesos proporcional a la frecuencia de aparición de cada categoría, garantizando que todas las etiquetas contribuyeran de manera más equilibrada al proceso de optimización. Los valores obtenidos para cada clase se resumen en la Tabla 8, donde puede observarse la diferencia de ponderación asignada a las categorías mayoritarias y minoritarias.

Figura 26. Obtención de la distribución de las etiquetas emocionales y cálculo de los pesos utilizados en la función de pérdida (Weighted Loss) mediante la función compute_class_weight. Elaboración propia.

Etiqueta	Peso clase 0	Peso clase 1
Vergüenza	0.979	1.022
Irritabilidad	0.995	1.005
Inseguridad	0.813	1.299
Depresión	1.203	0.856
Ira	0.776	1.405
Ansiedad	0.637	2.323

Tabla 8. Pesos calculados para cada categoría emocional utilizando la función compute_class_weight con la estrategia balanced. Elaboración propia.

Los pesos obtenidos fueron incorporados posteriormente a la función de pérdida BCEWithLogitsLoss, utilizada durante el entrenamiento de los modelos BERT, RoBERTa y DistilBERT. Como puede observarse en la Tabla 8, las categorías con menor número de ejemplos, como ansiedad, ira e inseguridad, recibieron una mayor ponderación durante el entrenamiento, mientras que las clases con mayor representación obtuvieron pesos relativamente menores. Esta estrategia permitió reducir el impacto del desbalance presente en el conjunto de datos y favorecer un aprendizaje más equilibrado, mejorando la capacidad de los modelos para identificar correctamente todas las etiquetas de la clasificación multietiqueta.

Entrenamiento del modelo BERT

Como primera arquitectura evaluada se seleccionó BERT (Bidirectional Encoder Representations from Transformers), utilizando el modelo preentrenado dccuchile/bert-base-spanish-wwm-cased, desarrollado específicamente para el idioma español. Este modelo fue elegido debido a su capacidad para generar representaciones contextuales bidireccionales del lenguaje, permitiendo interpretar

cada palabra considerando el contexto completo del comentario. Gracias a esta capacidad, BERT se ha consolidado como una de las arquitecturas más utilizadas en tareas de clasificación de textos y procesamiento del lenguaje natural, ofreciendo un alto desempeño en problemas de clasificación supervisada.

Con el propósito de adaptar el modelo al problema de clasificación multietiqueta planteado en esta investigación, se aplicó la técnica de Fine-Tuning, configurando una capa de salida con seis neuronas correspondientes a las categorías de vergüenza, irritabilidad, inseguridad, depresión, ira y ansiedad. Asimismo, se definió el tipo de problema como multi_label_classification, permitiendo que un mismo comentario pudiera ser clasificado simultáneamente en una o más categorías emocionales cuando así correspondiera. La configuración general utilizada para el entrenamiento del modelo se resume en la Tabla 9.

Parámetro	Valor
Modelo	dccuchile/bert-base-spanish-wwm-cased
Arquitectura	BERT Base
Tipo de problema	Clasificación multietiqueta
Número de etiquetas	6
Learning Rate	2×10^{-5}
Batch Size	8
Número de épocas	5
Weight Decay	0.01
Estrategia de evaluación	Por época
Mejor modelo	Sí (load_best_model_at_end=True)

Tabla 9. Configuración utilizada para el entrenamiento del modelo BERT. Elaboración propia.

Posteriormente, se configuraron los principales hiperparámetros del entrenamiento, incluyendo la tasa de aprendizaje, el tamaño del lote, el número de épocas y la estrategia de evaluación al finalizar cada ciclo de entrenamiento. Asimismo, se habilitó el almacenamiento automático del modelo con mejor desempeño sobre el conjunto de validación, garantizando la conservación de la configuración que obtuvo los mejores |resultados experimentales. La Figura 27 muestra la

implementación realizada para la carga del modelo preentrenado y la configuración de los parámetros empleados durante el proceso de ajuste fino.

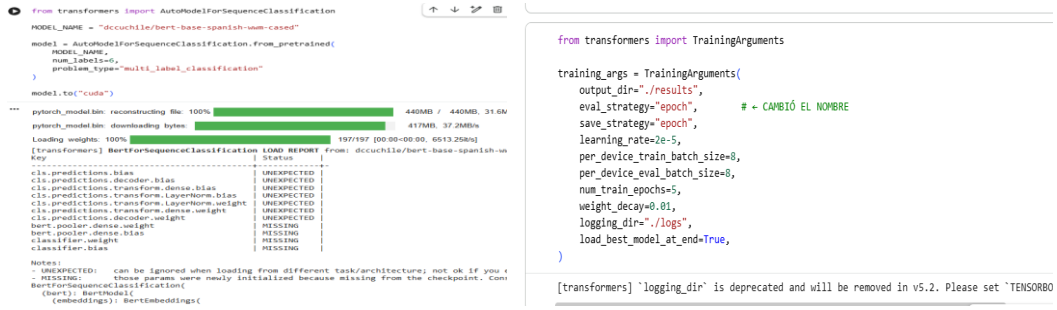


Figura 27. Carga del modelo preentrenado BERT y configuración de los parámetros utilizados durante el proceso de Fine-Tuning. Elaboración propia.

Durante el entrenamiento se monitoreó el comportamiento del modelo mediante el cálculo de la pérdida de entrenamiento (*Training Loss*), pérdida de validación (*Validation Loss*), exactitud (*Accuracy*), precisión (*Precisión*), exhaustividad (*Recall*) y medida F1 (*F1-score*). Estas métricas permitieron analizar la evolución del aprendizaje a lo largo de las cinco épocas y verificar la capacidad de generalización del modelo sobre datos no utilizados durante el entrenamiento. La Figura 28 presenta los resultados obtenidos durante el proceso de entrenamiento y la evaluación final realizada sobre el conjunto de prueba.

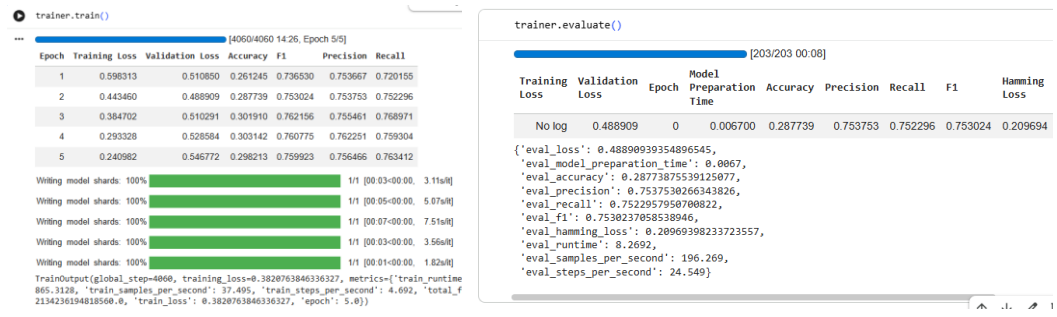


Figura 28. Proceso de entrenamiento y evaluación del modelo BERT durante las cinco épocas de Fine-Tuning. Elaboración propia.

Los resultados obtenidos evidenciaron un comportamiento estable durante el entrenamiento, alcanzando el mejor desempeño en la tercera época, con una exactitud de 0.3019, una precisión de 0.7555, un recall de 0.7690 y un F1-score de 0.7622. Aunque la pérdida de entrenamiento disminuyó progresivamente conforme avanzaron las épocas, se observó un ligero incremento en la pérdida de validación

durante las últimas épocas, indicando el inicio de un posible proceso de sobreajuste. En consecuencia, la tercera época fue considerada la configuración óptima para representar el desempeño del modelo BERT en la detección automática de indicadores lingüísticos de neuroticismo.

Evolución del entrenamiento durante las cinco épocas

Época	Training Loss	Validation Loss	Accuracy	Precision	Recall	F1-score
1	0.598313	0.510850	0.261245	0.753667	0.720155	0.736530
2	0.443460	0.488909	0.287739	0.753753	0.752296	0.753024
3	0.384702	0.510291	0.301910	0.755461	0.768971	0.762156
4	0.293328	0.528584	0.303142	0.762251	0.759304	0.760775
5	0.240982	0.546772	0.298213	0.756466	0.763412	0.759923

Tabla 10. Evolución del entrenamiento del modelo BERT durante las cinco épocas. Elaboración propia.

Durante las cinco épocas de entrenamiento se observó una disminución progresiva de la función de pérdida correspondiente al conjunto de entrenamiento (*Training Loss*), pasando de 0.5983 en la primera época a 0.2410 en la quinta. Este comportamiento evidencia que el modelo ajustó gradualmente sus parámetros internos conforme avanzó el proceso de aprendizaje, logrando una mejor representación de los patrones lingüísticos presentes en los comentarios utilizados para el entrenamiento. De manera paralela, la pérdida de validación presentó un comportamiento estable durante las primeras etapas y un ligero incremento en las últimas épocas, lo que sugiere el inicio de un posible proceso de sobreajuste sin afectar significativamente el desempeño general del modelo.

Asimismo, las métricas de evaluación mostraron una evolución favorable a lo largo del entrenamiento, alcanzando los mejores resultados durante la tercera época, con un Accuracy de 0.3019, una Precision de 0.7555, un Recall de 0.7690 y un F1-score de 0.7622. Estos resultados evidencian un equilibrio adecuado entre la capacidad del modelo para identificar correctamente los indicadores lingüísticos asociados al

neuroticismo y la reducción de errores de clasificación dentro del problema multietiqueta abordado en esta investigación.

Mejor desempeño obtenido por el modelo BERT

Métrica	Mejor valor
Accuracy	0.3019
Precisión	0.7555
Recall	0.7690
F1-score	0.7622
Mejor época	3

Tabla 11. Mejor desempeño obtenido por el modelo BERT durante el entrenamiento. Elaboración propia.

La Tabla 11 resume las mejores métricas alcanzadas por el modelo BERT durante el proceso de entrenamiento. Como puede observarse, el mayor rendimiento se obtuvo en la tercera época, donde se alcanzó el mejor equilibrio entre precisión, exhaustividad y medida F1. Estos resultados constituyen la primera referencia experimental de la investigación y servirán como base para comparar el desempeño de los modelos RoBERTa, DistilBERT y Flan-T5, permitiendo identificar las ventajas y limitaciones de cada arquitectura en la detección automática de indicadores lingüísticos asociados al rasgo de neuroticismo.

Entrenamiento del modelo RoBERTa

El segundo modelo evaluado durante la investigación fue RoBERTa (*Robustly Optimized BERT Pretraining Approach*), utilizando el modelo preentrenado bertin-project/bertin-roberta-base-spanish, desarrollado específicamente para el procesamiento de textos en español. Esta arquitectura constituye una versión optimizada de BERT, incorporando mejoras en la estrategia de preentrenamiento mediante el uso de un mayor volumen de datos, la eliminación de la tarea de predicción de la siguiente oración (*Next Sentence Prediction*) y una optimización más eficiente del proceso de aprendizaje. Gracias a estas características, RoBERTa ha demostrado un elevado desempeño en tareas de clasificación automática de textos y comprensión del lenguaje natural.

Al igual que en el modelo BERT, se aplicó la estrategia de Fine-Tuning para adaptar la arquitectura al problema de clasificación multietiqueta planteado en esta investigación. Para ello, el modelo fue configurado con una capa de salida compuesta por seis neuronas, correspondientes a las categorías de vergüenza, irritabilidad, inseguridad, depresión, ira y ansiedad, permitiendo que un mismo comentario pudiera asociarse simultáneamente a uno o varios indicadores emocionales. La configuración general utilizada durante el entrenamiento se resume en la Tabla 12.

Parámetro	Valor
Modelo preentrenado	bertin-project/bertin-roberta-base-spanish
Arquitectura	RoBERTa Base
Idioma	Español
Tipo de entrenamiento	Fine-Tuning
Tipo de clasificación	Multietiqueta
Número de etiquetas	6
Tokenizador	AutoTokenizer
Modelo	AutoModelForSequenceClassification
División del conjunto de datos	80 % entrenamiento – 20 % prueba
Registros de entrenamiento	6489
Registros de prueba	1623
Función de pérdida	BCEWithLogitsLoss con Weighted Loss
Optimizador	AdamW
Número de épocas	5
Plataforma	Google Colab
Framework	PyTorch + Hugging Face Transformers

Tabla 12. Configuración utilizada durante el entrenamiento del modelo RoBERTa. Elaboración propia.

Posteriormente, los comentarios fueron procesados mediante el tokenizador oficial de la arquitectura RoBERTa, convirtiendo cada texto en secuencias de identificadores numéricos y máscaras de atención necesarias para el procesamiento del lenguaje. El entrenamiento fue desarrollado utilizando la biblioteca Hugging

Face Transformers sobre el entorno de ejecución de Google Colab, manteniendo las mismas condiciones experimentales empleadas para el modelo BERT. Asimismo, se utilizó la función de pérdida BCEWithLogitsLoss con la estrategia de Weighted Loss, permitiendo compensar el desbalance existente entre las diferentes categorías emocionales. La Figura 29 presenta la implementación utilizada para la carga del modelo y la configuración de los parámetros empleados durante el proceso de ajuste fino.

Figura 29. Carga del modelo preentrenado RoBERTa y configuración de los parámetros. Elaboración propia.

Durante el entrenamiento se registró la evolución de la pérdida de entrenamiento (*Training Loss*), pérdida de validación (*Validation Loss*), exactitud (*Accuracy*), precisión (*Precisión*), exhaustividad (*Recall*) y medida F1 (*F1-score*) durante las cinco épocas de entrenamiento. Estas métricas permitieron evaluar el comportamiento del modelo y determinar la configuración con mejor capacidad de generalización sobre el conjunto de prueba. La Figura 30 muestra los resultados obtenidos durante el proceso de entrenamiento y la evaluación final del modelo.

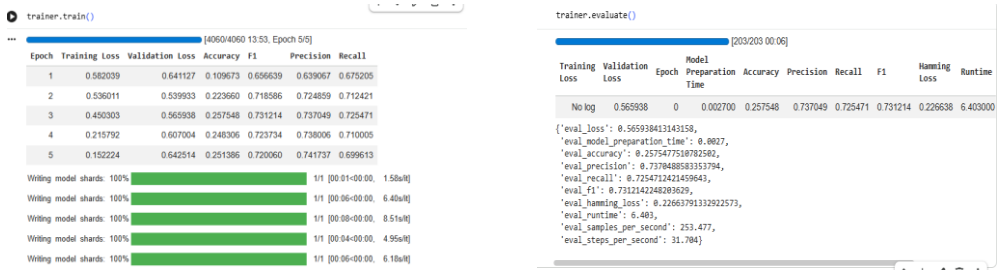


Figura 30. Proceso de entrenamiento y evaluación del modelo RoBERTa durante las épocas. Elaboración propia.

Época	Training Loss	Validation Loss	Accuracy	Precisión	Recall	F1-score
1	0.582039	0.641127	0.109673	0.639067	0.675205	0.656639
2	0.536011	0.539933	0.223660	0.724859	0.712421	0.718586
3	0.450303	0.565938	0.257548	0.737049	0.725471	0.731214
4	0.215792	0.607004	0.248306	0.738006	0.710005	0.723734
5	0.152224	0.642514	0.251386	0.741737	0.699613	0.720060

Tabla 13. Evolución del entrenamiento del modelo RoBERTa durante las cinco épocas. Elaboración propia.

Durante las cinco épocas de entrenamiento se observó una reducción progresiva de la función de pérdida correspondiente al conjunto de entrenamiento (*Training Loss*), disminuyendo de 0.5820 en la primera época a 0.1522 en la quinta. Este comportamiento evidencia que el modelo ajustó gradualmente sus parámetros internos conforme avanzó el proceso de aprendizaje, permitiéndole representar con mayor precisión los patrones lingüísticos presentes en los comentarios utilizados durante el entrenamiento. En cuanto a la pérdida de validación, el mejor resultado se obtuvo en la segunda época, con un valor de 0.5399, observándose posteriormente un ligero incremento que sugiere el inicio de un posible proceso de sobreajuste.

Asimismo, las métricas de evaluación alcanzaron su mejor desempeño durante la tercera época, registrando una exactitud de 0.2575, una precisión de 0.7370, un recall de 0.7255 y un F1-score de 0.7312. Estos resultados evidencian que el modelo logró mantener un equilibrio adecuado entre precisión y capacidad de recuperación

de las etiquetas emocionales, permitiendo identificar de forma satisfactoria los indicadores lingüísticos asociados al rasgo de neuroticismo.

Métrica	Mejor valor
Accuracy	0.2575
Precisión	0.7370
Recall	0.7255
F1-score	0.7312
Hamming Loss	0.2266
Mejor época	3

Tabla 14. Mejor desempeño obtenido por el modelo RoBERTa durante el entrenamiento. Elaboración propia.

La Tabla 14 resume las mejores métricas obtenidas por el modelo RoBERTa durante el proceso de entrenamiento. Aunque esta arquitectura presentó una reducción constante de la pérdida de entrenamiento y un desempeño estable durante las cinco épocas, los resultados obtenidos fueron ligeramente inferiores a los alcanzados por el modelo BERT en términos de F1-score. No obstante, RoBERTa demostró una adecuada capacidad para identificar los indicadores lingüísticos asociados al neuroticismo, constituyendo una referencia importante para comparar posteriormente su comportamiento con los modelos DistilBERT y Flan-T5 implementados en la presente investigación.

Entrenamiento del modelo DistilBERT

El tercer modelo evaluado durante la investigación fue DistilBERT, utilizando la versión preentrenada dccuchile/distilbert-base-spanish-uncased, desarrollada específicamente para el procesamiento de textos en español. Esta arquitectura corresponde a una versión optimizada de BERT obtenida mediante técnicas de Knowledge Distillation, cuyo objetivo consiste en reducir el tamaño del modelo y el costo computacional sin afectar significativamente su capacidad de comprensión del lenguaje. Gracias a estas características, DistilBERT constituye una alternativa eficiente para tareas de procesamiento del lenguaje natural que requieren tiempos de entrenamiento e inferencia menores, manteniendo un desempeño competitivo respecto a arquitecturas Transformer de mayor complejidad.

Al igual que en los modelos previamente evaluados, se aplicó la estrategia de Fine-Tuning para adaptar DistilBERT al problema de clasificación multietiqueta planteado en esta investigación. Para ello, el modelo fue configurado con una capa de salida compuesta por seis neuronas, correspondientes a las categorías de vergüenza, irritabilidad, inseguridad, depresión, ira y ansiedad, permitiendo identificar simultáneamente uno o varios indicadores emocionales presentes en cada comentario. La configuración general utilizada durante el entrenamiento del modelo se presenta en la Tabla 15.

Parámetro	Valor Tabla
Modelo	dccuchile/distilbert-base-spanish-uncased
Arquitectura	DistilBERT
Tipo de problema	Clasificación multietiqueta
Número de etiquetas	6
Learning Rate	2×10^{-5}
Batch Size	8
Número de épocas	5
Weight Decay	0.01
Estrategia de evaluación	Por época
Mejor modelo	Sí (load_best_model_at_end=True)

Tabla 15. Configuración utilizada para el entrenamiento del modelo DistilBERT. Elaboración propia.

Posteriormente, los comentarios fueron procesados mediante el tokenizador oficial de DistilBERT, convirtiendo cada texto en secuencias de identificadores numéricos y máscaras de atención necesarias para el procesamiento del lenguaje. El entrenamiento fue desarrollado utilizando la biblioteca Hugging Face Transformers sobre el entorno de ejecución de Google Colab, manteniendo las mismas condiciones experimentales empleadas para los modelos BERT y RoBERTa. Asimismo, se utilizó la función de pérdida BCEWithLogitsLoss con la estrategia de Weighted Loss, permitiendo compensar el desbalance existente entre las diferentes categorías emocionales. La Figura 31 muestra la implementación

utilizada para la carga del modelo y la configuración de los parámetros empleados durante el proceso de ajuste fino.

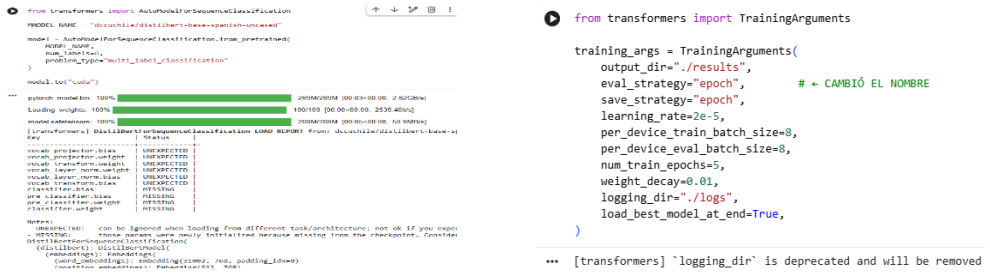


Figura 31. Carga del modelo preentrenado DistilBERT y configuración de los parámetros. Elaboración propia.

Durante el entrenamiento se registró la evolución de la pérdida de entrenamiento (*Training Loss*), pérdida de validación (*Validation Loss*), exactitud (*Accuracy*), precisión (*Precisión*), exhaustividad (*Recall*) y medida F1 (*F1-score*) durante las cinco épocas de entrenamiento. Estas métricas permitieron evaluar el comportamiento del modelo y determinar la configuración con mejor capacidad de generalización sobre el conjunto de prueba. La Figura 32 presenta los resultados obtenidos durante el proceso de entrenamiento y la evaluación final del modelo.

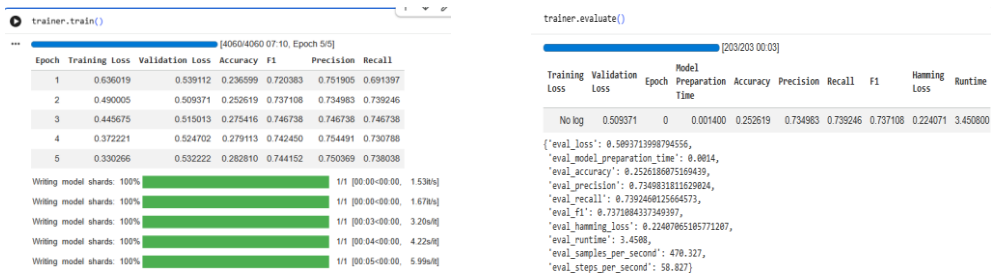


Figura 32. Proceso de entrenamiento y evaluación del modelo DistilBERT durante las épocas. Elaboración propia.

Época	Training Loss	Validation Loss	Accuracy	Precisión	Recall	F1-score
1	0.636019	0.539112	0.236599	0.751905	0.691397	0.720383
2	0.490005	0.509371	0.252619	0.734983	0.739246	0.737108
3	0.445675	0.515013	0.275416	0.746738	0.746738	0.746738
4	0.372221	0.524702	0.279113	0.754491	0.730788	0.742450
5	0.330266	0.532222	0.282810	0.750369	0.738038	0.744152

Tabla 16. Evolución del entrenamiento del modelo DistilBERT durante las cinco épocas. Elaboración propia.

Durante las cinco épocas de entrenamiento se observó una disminución progresiva de la pérdida correspondiente al conjunto de entrenamiento (*Training Loss*), pasando de 0.6360 en la primera época a 0.3303 en la quinta. Este comportamiento evidencia que el modelo ajustó gradualmente sus parámetros internos conforme avanzó el proceso de aprendizaje, logrando representar con mayor precisión los patrones lingüísticos presentes en los comentarios utilizados durante el entrenamiento. En cuanto a la pérdida de validación, el mejor resultado se obtuvo en la segunda época, con un valor de 0.5094, observándose posteriormente un ligero incremento durante las últimas etapas del entrenamiento, lo que sugiere el inicio de un posible proceso de sobreajuste.

Asimismo, las métricas de evaluación alcanzaron su mejor desempeño durante la tercera época, registrando una exactitud de 0.2754, una precisión de 0.7467, un recall de 0.7467 y un F1-score de 0.7467. Estos resultados evidencian que DistilBERT logró mantener un equilibrio adecuado entre precisión y capacidad de recuperación de las etiquetas emocionales, obteniendo un desempeño competitivo a pesar de tratarse de una arquitectura más ligera y eficiente desde el punto de vista computacional.

Métrica	Mejor valor
Accuracy	0.2754
Precisión	0.7467
Recall	0.7467
F1-score	0.7467
Hamming Loss	0.2241
Mejor época	3

Tabla 17. Mejor desempeño obtenido por el modelo DistilBERT durante el entrenamiento. Elaboración propia.

La Tabla 17 resume las mejores métricas obtenidas por el modelo DistilBERT durante el proceso de entrenamiento. Aunque esta arquitectura posee un menor número de parámetros que BERT y RoBERTa, logró alcanzar un desempeño

competitivo en la clasificación de los indicadores lingüísticos asociados al neuroticismo. Su F1-score de 0.7467 fue superior al obtenido por RoBERTa e inferior al alcanzado por BERT, evidenciando que DistilBERT representa una alternativa eficiente al ofrecer un equilibrio entre capacidad predictiva y costo computacional. Estos resultados constituyen la tercera referencia experimental de la investigación y servirán como base para comparar posteriormente el comportamiento del modelo Flan-T5 dentro del enfoque de *Few-Shot Learning*.

Implementación del modelo Flan-T5 mediante aprendizaje Few-Shot

El cuarto modelo evaluado durante la investigación fue FLAN-T5, una arquitectura de tipo *Encoder–Decoder* previamente ajustada mediante *Instruction Tuning*. A diferencia de BERT, RoBERTa y DistilBERT, este modelo no fue sometido a un proceso de *Fine-Tuning* sobre el conjunto completo de entrenamiento, sino que se implementó mediante una estrategia de *Few-Shot Learning*. Este enfoque permitió proporcionar al modelo instrucciones y ejemplos representativos dentro del *prompt*, con el propósito de orientar la identificación de los seis indicadores lingüísticos considerados en la investigación: vergüenza, irritabilidad, inseguridad, depresión, ira y ansiedad.

Parámetro	Valor
Modelo preentrenado	google/flan-t5-base
Arquitectura	Flan-T5 Base
Idioma de los textos analizados	Español
Estrategia utilizada	Few-Shot Learning
Tipo de clasificación	Multietiqueta
Número de etiquetas	6
Tokenizador	AutoTokenizer
Modelo	AutoModelForSeq2SeqLM
Método de inferencia	Prompt Engineering
Ajuste de parámetros del modelo	No aplica
Plataforma	Google Colab

Parámetro	Valor
Framework	Hugging Face Transformers

Tabla 18. Configuración utilizada para la implementación del modelo Flan-T5. Elaboración propia.

La implementación de FLAN-T5 se realizó de manera independiente al procedimiento de división 80/20 empleado para los modelos ajustados mediante *Fine-Tuning*. Para evitar que los mismos comentarios utilizados durante la configuración del enfoque fueran empleados posteriormente en la evaluación final, los datos correspondientes a FLAN-T5 se organizaron en tres grupos con funciones diferentes: ejemplos incluidos en el *prompt*, conjunto de Desarrollo y conjunto de Pruebas. Esta separación permitió mantener los comentarios destinados a la evaluación final independientes de aquellos empleados durante la configuración del procedimiento de clasificación.

Conjunto	**Cantidad**	**Finalidad**
Ejemplos del prompt	13	Proporcionar demostraciones Few-Shot al modelo
Desarrollo	50	Configurar y analizar el procedimiento de decisión
Pruebas	120	Realizar la evaluación final sobre comentarios independientes

Tabla 19. Organización de los conjuntos utilizados para la implementación de FLAN-T5 mediante Few-Shot Learning. Elaboración propia.

Durante la construcción del *prompt* se incorporaron instrucciones orientadas a la tarea de clasificación multietiqueta, junto con ejemplos que permitieron proporcionar al modelo información contextual sobre las categorías consideradas. Debido a las restricciones asociadas a la longitud máxima de entrada del modelo, se controló la cantidad de información incluida en el *prompt* para evitar superar el límite establecido durante la tokenización. De esta manera, se buscó mantener un equilibrio entre la cantidad de ejemplos proporcionados y el espacio disponible para procesar cada comentario nuevo.

A diferencia de los modelos de clasificación ajustados mediante *Fine-Tuning*, FLAN-T5 no genera directamente un vector de probabilidades mediante una capa de clasificación específica para las seis etiquetas. Por esta razón, la implementación Few-Shot se estructuró para obtener una puntuación para cada indicador a partir del comportamiento del modelo frente a las instrucciones proporcionadas. Las puntuaciones obtenidas fueron posteriormente contrastadas con los criterios de decisión establecidos durante la etapa de Desarrollo, permitiendo determinar la presencia o ausencia de cada indicador lingüístico en el comentario analizado.

Configuración mediante el conjunto de Desarrollo

El conjunto de Desarrollo estuvo compuesto por 50 comentarios y fue utilizado exclusivamente para analizar el comportamiento de las puntuaciones generadas por FLAN-T5 y establecer los criterios de decisión antes de realizar la evaluación final. Dentro de este conjunto se incluyeron comentarios que presentaban uno o varios indicadores lingüísticos y comentarios sin presencia de las categorías evaluadas, permitiendo observar el comportamiento del modelo en ambos escenarios.

Durante esta etapa se analizó la probabilidad máxima generada para cada comentario y se evaluaron diferentes valores de decisión para distinguir los textos que presentaban algún indicador de aquellos clasificados como ausencia de rasgos. El análisis permitió seleccionar un umbral global de abstención de 0.19, con el cual se obtuvo una *Balanced Accuracy* de 0.7250, un F1 macro de 0.7159 para la detección de presencia o ausencia, una precisión de 0.8077 y un recall de 0.7000 sobre el conjunto de Desarrollo. Estos valores fueron utilizados únicamente para configurar el procedimiento antes de aplicar la evaluación sobre el conjunto independiente de Pruebas.

Parámetro/Métrica	Valor
Comentarios de Desarrollo	50
Umbral global de abstención	0.19
Balanced Accuracy	0.7250
F1 macro presencia/ausencia	0.7159

Parámetro/Métrica	Valor
Precisión de presencia	0.8077
Recall de presencia	0.7000

Tabla 20. Resultados obtenidos durante la configuración del enfoque Few-Shot sobre el conjunto de Desarrollo. Elaboración propia.

Una vez definidos los criterios de decisión mediante el conjunto de Desarrollo, estos se mantuvieron sin modificaciones durante la evaluación final. De esta forma, los 120 comentarios correspondientes al conjunto de Pruebas no fueron utilizados para seleccionar umbrales ni realizar ajustes posteriores, permitiendo obtener una valoración más representativa de la capacidad de generalización del enfoque implementado.

Evaluación final de FLAN-T5

La evaluación final se realizó sobre 120 comentarios independientes, comparando para cada registro las etiquetas reales con las predicciones obtenidas mediante FLAN-T5. Al tratarse de un problema multietiqueta, se calcularon métricas de evaluación tanto a nivel micro como macro, además de *Accuracy* mediante coincidencia exacta y *Hamming Loss*. Este procedimiento permitió analizar el comportamiento general del modelo y las diferencias existentes en su capacidad para identificar cada uno de los seis indicadores lingüísticos.

Los resultados muestran que FLAN-T5 obtuvo un Accuracy de 0.1583, indicando que el 15,83 % de los comentarios presentó una coincidencia exacta entre el conjunto completo de etiquetas reales y predichas. En la evaluación micro se alcanzó una precisión de 0.4679, un recall de 0.1802 y un F1-score de 0.2602. Por otra parte, las métricas macro registraron una precisión de 0.4080, un recall de 0.2060 y un F1-score de 0.2484. Finalmente, el *Hamming Loss* alcanzó un valor de 0.4028.

Métrica	Valor
Accuracy (Exact Match)	0.1583

Métrica	Valor
Precision micro	0.4679
Recall micro	0.1802
F1-score micro	0.2602
Precision macro	0.4080
Recall macro	0.2060
F1-score macro	0.2484
Hamming Loss	0.4028

Tabla 21. Métricas obtenidas por FLAN-T5 sobre los 120 comentarios del conjunto de Pruebas. Elaboración propia

El análisis por indicador evidenció diferencias importantes en la capacidad de detección del modelo. La categoría inseguridad alcanzó un F1-score de 0.4722, mientras que ira obtuvo 0.3824, depresión 0.2963, ansiedad 0.2500 y vergüenza 0.0896. En el caso de irritabilidad, el modelo no logró identificar correctamente casos positivos en el conjunto de Pruebas, obteniendo un F1-score de 0.0000. Estos resultados muestran que el comportamiento del enfoque Few-Shot no fue uniforme entre las seis categorías evaluadas.

Indicador	Precisión	Recall	F1-score	Soporte
Vergüenza	0.6000	0.0484	0.0896	62
Irritabilidad	0.0000	0.0000	0.0000	56
Inseguridad	0.6071	0.3864	0.4722	44
Depresión	0.5714	0.2000	0.2963	60
Ira	0.4194	0.3514	0.3824	37
Ansiedad	0.2500	0.2500	0.2500	24

Tabla 22. Resultados obtenidos por FLAN-T5 para cada indicador lingüístico en el conjunto de Pruebas. Elaboración propia.

Los resultados obtenidos permiten observar que el enfoque Few-Shot presentó una capacidad limitada para generalizar sobre los comentarios independientes utilizados en la evaluación final, especialmente en determinadas categorías. El bajo recall global indica que una proporción importante de los indicadores presentes en los

textos no fue identificada por el modelo. No obstante, el experimento permitió evaluar una estrategia diferente al *Fine-Tuning* y evidenciar el comportamiento de un modelo basado en instrucciones cuando dispone de una cantidad reducida de ejemplos dentro del *prompt*.

La separación establecida entre los ejemplos utilizados en el *prompt*, el conjunto de Desarrollo y el conjunto de Pruebas permitió mantener una evaluación independiente del enfoque Few-Shot implementado. Los criterios de decisión fueron definidos previamente mediante el conjunto de Desarrollo y posteriormente aplicados, sin modificaciones, sobre los 120 comentarios reservados para la evaluación final. De esta manera, los resultados obtenidos reflejan con mayor objetividad el comportamiento de FLAN-T5 frente a textos no utilizados durante la configuración del procedimiento y permiten realizar posteriormente una comparación con los modelos ajustados mediante *Fine-Tuning*.

3.4. Resultados

Los resultados obtenidos durante la presente investigación permitieron analizar el desempeño de los modelos BERT, RoBERTa, DistilBERT y FLAN-T5 para la detección automática de indicadores lingüísticos asociados al neuroticismo en textos escritos en español. Para la evaluación se utilizaron métricas como Accuracy, Precision, Recall, F1-score y Hamming Loss, considerando las características particulares del procedimiento experimental aplicado a cada arquitectura.

Los modelos BERT, RoBERTa y DistilBERT fueron ajustados mediante Fine-Tuning y evaluados sobre el conjunto de prueba definido a partir de la división 80/20. Por su parte, FLAN-T5 fue implementado mediante Few-Shot Learning y evaluado sobre un conjunto independiente de 120 comentarios, después de establecer los criterios de decisión mediante el conjunto de Desarrollo. Debido a estas diferencias metodológicas, los resultados de FLAN-T5 se presentan como referencia experimental del enfoque Few-Shot y no como una comparación directa bajo las mismas condiciones de entrenamiento de los modelos ajustados mediante Fine-Tuning. La Tabla 23 resume el desempeño alcanzado por cada una de las arquitecturas implementadas.

Modelo	Estrategia	Accuracy	Precisión	Recall	F1-score	Hamming Loss
BERT	Fine-Tuning	0.3019	0.7555	0.7690	**0.7622**	0.2097
RoBERTa	Fine-Tuning	0.2575	0.7370	0.7255	0.7312	0.2266
DistilBERT	Fine-Tuning	0.2754	0.7467	0.7467	0.7467	0.2241
FLAN-T5	Few-Shot	0.1583	0.4679	0.1802	0.2602	0.4028

Tabla 23. Comparación del desempeño de los modelos implementados. Elaboración propia.

Nota: Para FLAN-T5 se presentan los valores de Precision, Recall y F1-score correspondientes al promedio micro. Los resultados fueron obtenidos mediante *Few-Shot Learning* sobre un conjunto independiente de 120 comentarios, por lo que deben interpretarse considerando las diferencias metodológicas respecto a los modelos ajustados mediante *Fine-Tuning*.

Comparación del desempeño de los modelos

La comparación de los resultados evidencia diferencias en el desempeño de las arquitecturas implementadas y en las estrategias utilizadas durante la investigación. Entre los modelos ajustados mediante Fine-Tuning, BERT alcanzó el mejor rendimiento general con un F1-score de 0.7622, una precisión de 0.7555 y un recall de 0.7690. Estos resultados muestran un equilibrio adecuado entre la identificación de los indicadores presentes en los textos y la capacidad del modelo para reducir los errores de clasificación.

DistilBERT obtuvo un F1-score de 0.7467, manteniendo un rendimiento cercano al alcanzado por BERT pese a utilizar una arquitectura más compacta. Por su parte, RoBERTa alcanzó un F1-score de 0.7312, presentando también un comportamiento

competitivo dentro de la tarea de clasificación multietiqueta. En conjunto, los tres modelos ajustados mediante Fine-Tuning mostraron una mayor capacidad para adaptarse a las características del conjunto de datos utilizado en la investigación.

En el caso de FLAN-T5, la evaluación mediante Few-Shot Learning produjo un F1-score micro de 0.2602, una precisión micro de 0.4679 y un recall micro de 0.1802. El valor reducido de recall evidencia dificultades para identificar una parte importante de los indicadores presentes en los comentarios del conjunto de Pruebas. Sin embargo, estos resultados corresponden a una estrategia diferente, en la cual el modelo no fue sometido a Fine-Tuning y utilizó un número reducido de ejemplos dentro del prompt para orientar la clasificación.

Los resultados obtenidos permiten observar que, bajo las condiciones experimentales establecidas en esta investigación, el ajuste mediante Fine-Tuning presentó un desempeño superior al enfoque Few-Shot implementado con FLAN-T5. Esta diferencia resulta especialmente visible en el F1-score y el recall obtenidos durante la evaluación. No obstante, la incorporación de FLAN-T5 permitió explorar una alternativa basada en aprendizaje con pocos ejemplos y analizar sus posibilidades y limitaciones dentro de la detección multietiqueta de indicadores lingüísticos asociados al neuroticismo.

Selección del modelo final

A partir de los resultados obtenidos durante la evaluación experimental, se seleccionó BERT como modelo principal para la detección automática de indicadores lingüísticos asociados al neuroticismo en la aplicación desarrollada. Entre las arquitecturas ajustadas mediante *Fine-Tuning*, este modelo alcanzó el mayor F1-score con un valor de 0.7622, acompañado de una precisión de 0.7555 y un recall de 0.7690. Estos resultados evidenciaron un comportamiento equilibrado en la identificación de las diferentes categorías consideradas dentro de la tarea de clasificación multietiqueta.

Aunque DistilBERT y RoBERTa alcanzaron resultados competitivos, con F1-score de 0.7467 y 0.7312 respectivamente, BERT presentó el mejor desempeño global bajo las condiciones experimentales establecidas. Este comportamiento permitió

seleccionarlo como la arquitectura principal para procesar los comentarios ingresados por los usuarios y generar las probabilidades correspondientes a los seis indicadores lingüísticos evaluados.

Por otra parte, FLAN-T5 se mantuvo en la propuesta como una alternativa experimental basada en *Few-Shot Learning*. Su evaluación independiente evidenció un rendimiento inferior al alcanzado por los modelos ajustados mediante *Fine-Tuning*, con un F1-score micro de 0.2602. Debido a las diferencias existentes entre ambas estrategias, FLAN-T5 no fue seleccionado como modelo principal, pero su implementación permitió analizar el comportamiento de un enfoque basado en instrucciones y pocos ejemplos frente a la clasificación multietiqueta planteada.

En consecuencia, la aplicación utiliza BERT como modelo principal para realizar el análisis de los comentarios y presentar los indicadores detectados junto con sus respectivas probabilidades. FLAN-T5 se incorpora de manera independiente como una funcionalidad experimental que permite ejecutar el análisis mediante *Few-Shot Learning*, manteniendo así las dos estrategias evaluadas durante el desarrollo de la investigación.

3.5. Pruebas

Las pruebas funcionales se realizaron con el propósito de verificar el funcionamiento de la aplicación desarrollada para la detección automática de indicadores lingüísticos asociados al neuroticismo en textos escritos en español. Para ello, se evaluó el ingreso de comentarios, la ejecución del proceso de clasificación mediante el modelo BERT y la presentación de los resultados correspondientes a los seis indicadores considerados en la investigación. Asimismo, se comprobó el comportamiento de la aplicación ante comentarios en los que no se identificaron indicadores, evitando generar una clasificación cuando las probabilidades obtenidas no superaron los criterios establecidos.

Adicionalmente, se verificó de manera independiente la funcionalidad correspondiente a FLAN-T5, implementada mediante *Few-Shot Learning*. Esta opción permite analizar un comentario utilizando el enfoque basado en instrucciones y ejemplos desarrollado durante la investigación, manteniéndolo

separado del análisis principal realizado mediante BERT. Las pruebas permitieron comprobar la integración de ambos enfoques dentro de la aplicación y verificar que los resultados fueran presentados de acuerdo con las características de cada modelo.

3.5.1. Prueba de ingreso del comentario

La primera prueba funcional tuvo como objetivo verificar el correcto ingreso de los comentarios que serán procesados por la aplicación. Para ello, se introdujeron textos escritos en español en el campo habilitado para el análisis y se comprobó que el sistema permitiera registrar correctamente la información antes de ejecutar el proceso de clasificación. Asimismo, se verificó el funcionamiento de las opciones destinadas a iniciar el análisis y limpiar el contenido ingresado.

Durante la prueba se comprobó que la aplicación mantuvo correctamente el comentario ingresado y permitió ejecutar las acciones disponibles sin presentar errores de funcionamiento. También se verificó el control de entradas vacías, evitando iniciar el procesamiento cuando no existe información suficiente para realizar el análisis. Estos resultados permitieron comprobar el funcionamiento adecuado del mecanismo de entrada de información utilizado por la aplicación. La Figura 33Figura 33. Prueba de ingreso del comentario en la aplicación. Elaboración propia. presenta la ejecución de esta prueba.

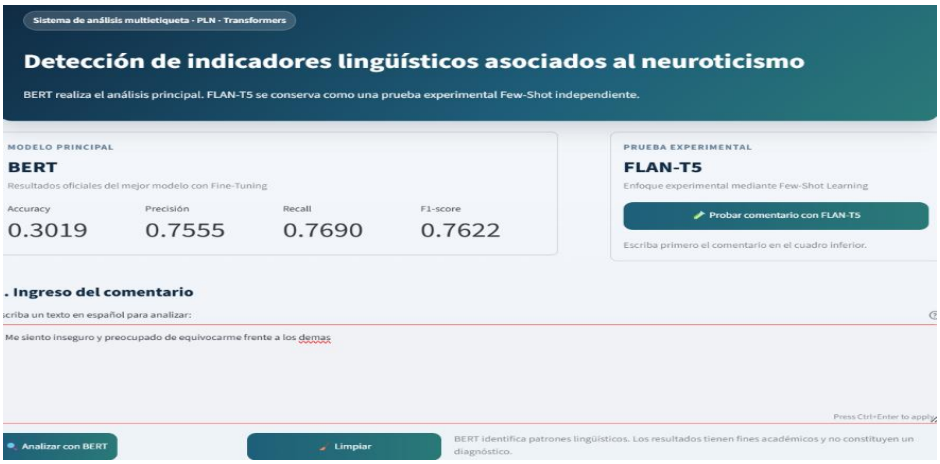


Figura 33. Prueba de ingreso del comentario en la aplicación. Elaboración propia.

3.5.2. Prueba de clasificación automática con BERT

La segunda prueba tuvo como finalidad verificar el proceso de clasificación automática utilizando BERT, seleccionado como modelo principal de la aplicación a partir de los resultados obtenidos durante la evaluación experimental. Para realizar la prueba se ingresó un comentario en español y se ejecutó la opción correspondiente al análisis, permitiendo que el texto fuera procesado y enviado al modelo para obtener las probabilidades asociadas a los seis indicadores lingüísticos considerados.

Como resultado, la aplicación presentó los indicadores detectados junto con las probabilidades generadas por BERT, permitiendo identificar las categorías con mayor presencia dentro del comentario analizado. La información obtenida fue complementada mediante una representación gráfica tipo radar, en la cual se muestran los valores correspondientes a vergüenza, irritabilidad, inseguridad, depresión, ira y ansiedad. Esta representación facilita la interpretación del comportamiento del modelo frente al texto proporcionado por el usuario. La Figura 34 presenta el resultado obtenido durante la clasificación.

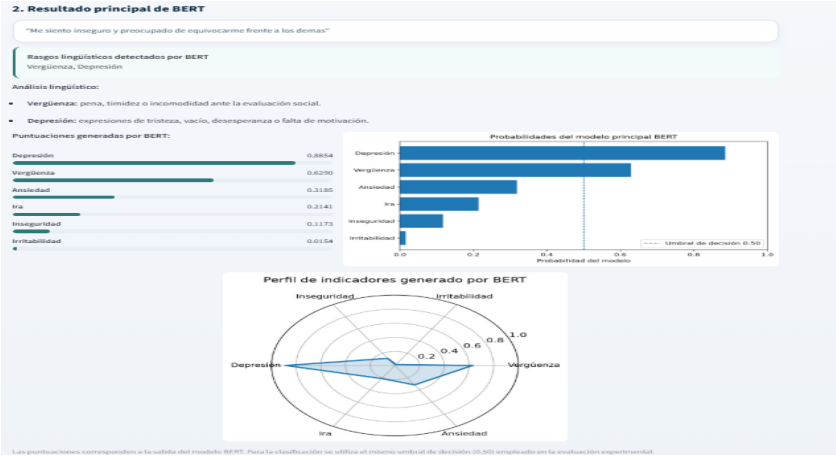


Figura 34. Prueba del proceso de clasificación automática del comentario. Elaboración propia.

También se verificó el comportamiento de la aplicación frente a comentarios en los que las probabilidades obtenidas no permiten identificar indicadores lingüísticos asociados al neuroticismo. En estos casos, el sistema presenta un mensaje indicando que no se detectaron indicadores, evitando mostrar una clasificación positiva

cuando no se cumplen los criterios definidos para la detección. Esta prueba permitió comprobar el control implementado para los comentarios que no presentan indicadores detectables. Figura 35 muestra el resultado correspondiente a este escenario.

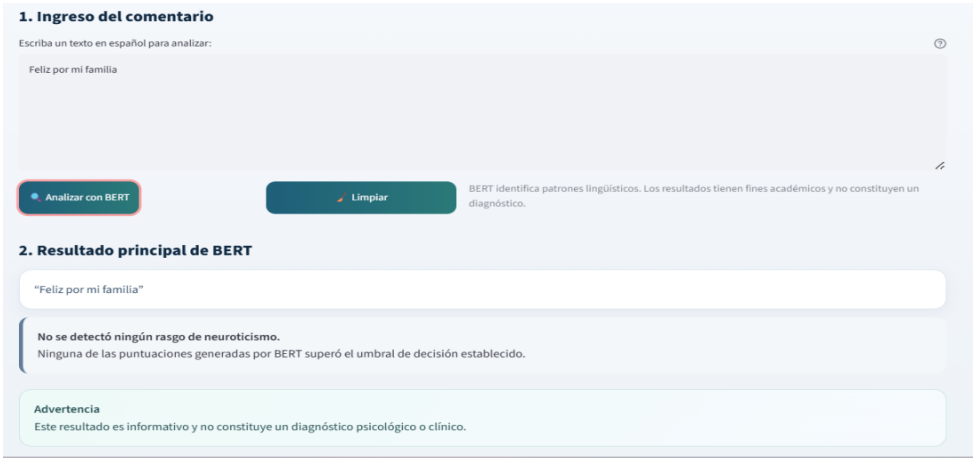


Figura 35. Prueba de clasificación de un comentario sin indicadores lingüísticos detectados. Elaboración propia.

3.5.3. Prueba de FLAN-T5 mediante Few-Shot Learning

La tercera prueba estuvo orientada a verificar la funcionalidad experimental correspondiente a FLAN-T5 mediante *Few-Shot Learning*. A diferencia del análisis principal realizado con BERT, esta opción ejecuta de manera independiente el procedimiento basado en instrucciones y ejemplos definido para FLAN-T5, permitiendo analizar un comentario sin realizar un proceso adicional de *Fine-Tuning* sobre el modelo.

Para la ejecución de la prueba se ingresó un comentario en español y se seleccionó la opción correspondiente a FLAN-T5. El sistema procesó los indicadores lingüísticos considerados y aplicó los criterios de decisión establecidos previamente mediante el conjunto de Desarrollo. De esta manera, la aplicación presentó los indicadores identificados por el enfoque *Few-Shot* o, cuando no se alcanzaron los criterios establecidos, informó que no se detectaron indicadores en el comentario analizado.

La incorporación de esta funcionalidad permite mostrar dentro de la aplicación el segundo enfoque experimental considerado en la investigación sin establecer una comparación directa con BERT, debido a que ambos modelos fueron implementados mediante procedimientos diferentes. La prueba permitió comprobar la ejecución independiente de FLAN-T5 y la presentación de sus resultados dentro de la herramienta desarrollada. La Figura 36 presenta el resultado de la prueba realizada.

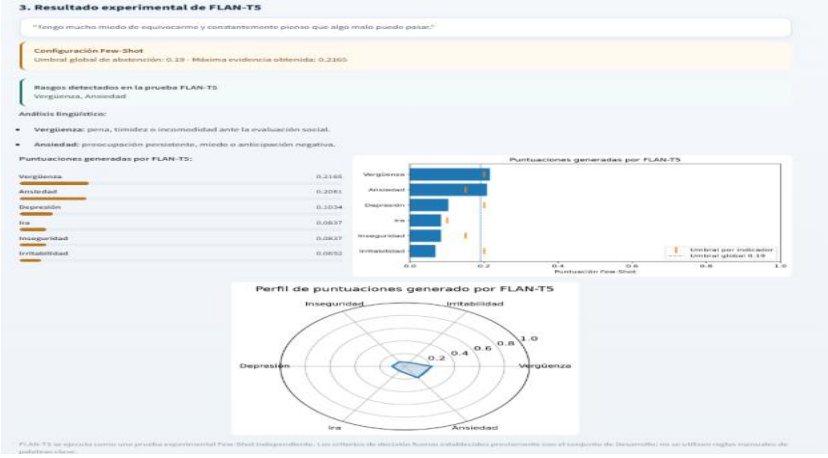


Figura 36. Prueba de análisis de un comentario mediante FLAN-T5 utilizando Few-Shot Learning. Elaboración propia.

Las pruebas funcionales realizadas permitieron comprobar el funcionamiento de los principales componentes de la aplicación desarrollada. El sistema permitió ingresar comentarios, ejecutar la clasificación mediante BERT como modelo principal, representar gráficamente los resultados y controlar los casos en los que no se detectaron indicadores lingüísticos. Asimismo, se verificó la ejecución independiente de FLAN-T5 mediante *Few-Shot Learning*, manteniendo esta funcionalidad como parte del análisis experimental desarrollado durante la investigación.

CONCLUSIONES

La presente investigación tuvo como objetivo desarrollar un sistema para la detección automática de indicadores lingüísticos asociados al rasgo de neuroticismo en textos escritos en español mediante técnicas de aprendizaje profundo. A partir del procesamiento del conjunto de datos, la implementación de diferentes arquitecturas Transformer, la evaluación de estrategias de *Fine-Tuning* y *Few-Shot Learning* y el desarrollo de una aplicación para la ejecución de los modelos, se establecen las siguientes conclusiones:

- Se logró preparar un conjunto de datos orientado a la detección automática de seis indicadores lingüísticos asociados al neuroticismo: vergüenza, irritabilidad, inseguridad, depresión, ira y ansiedad. El proceso incluyó la revisión y reetiquetado de los comentarios, así como etapas de normalización y preprocesamiento para adecuar los textos a los requerimientos de los modelos implementados. La naturaleza multietiqueta y el desbalance existente entre las categorías fueron considerados durante el desarrollo experimental, permitiendo establecer un procedimiento acorde con las características de los datos analizados.
- La evaluación de BERT, RoBERTa y DistilBERT mediante *Fine-Tuning* permitió determinar que BERT presentó el mejor desempeño entre estas arquitecturas, alcanzando un F1-score de 0.7622, una precisión de 0.7555 y un recall de 0.7690. DistilBERT obtuvo un F1-score de 0.7467, mientras que RoBERTa alcanzó 0.7312. Estos resultados evidencian que las arquitecturas Transformer ajustadas al conjunto de datos lograron adaptarse a la tarea de clasificación multietiqueta planteada y sustentaron la selección de BERT como modelo principal de la aplicación.
- La implementación de FLAN-T5 mediante *Few-Shot Learning* permitió evaluar una estrategia diferente al ajuste mediante *Fine-Tuning*. La evaluación final sobre 120 comentarios independientes obtuvo un Accuracy de 0.1583, una precisión micro de 0.4679, un recall micro de 0.1802 y un F1-score micro de 0.2602, evidenciando dificultades para generalizar e identificar una proporción importante de los indicadores presentes en textos

no utilizados durante la configuración del procedimiento. Estos resultados muestran que, bajo las condiciones experimentales de la investigación, el enfoque *Few-Shot* implementado con FLAN-T5 presentó un rendimiento inferior al alcanzado por las arquitecturas ajustadas mediante *Fine-Tuning*.

- Se desarrolló una aplicación que integra BERT como modelo principal para el análisis automático de comentarios y la presentación de los indicadores lingüísticos detectados junto con sus respectivas probabilidades. La herramienta incorpora una representación gráfica que facilita la interpretación de los resultados y controla los casos en los que no se identifican indicadores. Adicionalmente, FLAN-T5 se integró como una funcionalidad experimental independiente basada en *Few-Shot Learning*. Las pruebas funcionales permitieron verificar la ejecución de ambos enfoques y su integración dentro de la propuesta desarrollada.
- Los resultados obtenidos permitieron cumplir con el propósito de la investigación al desarrollar y evaluar una propuesta basada en aprendizaje profundo para la detección automática de indicadores lingüísticos asociados al neuroticismo en textos escritos en español. La experimentación evidenció que el desempeño depende tanto de la arquitectura seleccionada como de la estrategia utilizada para adaptarla a la tarea específica, destacándose el *Fine-Tuning* de BERT como la alternativa con mejor comportamiento dentro de las condiciones establecidas en el estudio.

RECOMENDACIONES

Con base en los resultados obtenidos y las limitaciones identificadas durante el desarrollo de la investigación, se plantean las siguientes recomendaciones para futuros trabajos relacionados con la detección automática de indicadores lingüísticos asociados al neuroticismo:

- Ampliar y diversificar el conjunto de datos mediante la incorporación de textos provenientes de diferentes plataformas y contextos comunicativos en español. Asimismo, se recomienda incrementar la cantidad de ejemplos correspondientes a las categorías con menor representación, con el propósito de reducir el efecto del desbalance y favorecer una mayor capacidad de generalización de los modelos.
- Explorar estrategias adicionales de optimización para BERT, RoBERTa y DistilBERT mediante ajuste de hiperparámetros, diferentes funciones de pérdida y procedimientos de validación que permitan analizar posibles mejoras en la detección de los seis indicadores lingüísticos, particularmente en aquellas categorías que presentan mayor dificultad de clasificación.
- Profundizar en la implementación de FLAN-T5 y otros modelos basados en instrucciones mediante la utilización de diferentes configuraciones de *prompts*, una mayor cantidad y diversidad de ejemplos representativos y otras estrategias de aprendizaje con pocos ejemplos. Los resultados obtenidos evidenciaron limitaciones en la capacidad de generalización del enfoque *Few-Shot* implementado, por lo que futuras investigaciones podrían analizar configuraciones que permitan incrementar especialmente el recall y el F1-score.
- Evaluar arquitecturas Transformer y modelos de lenguaje adicionales que dispongan de una mayor capacidad para el procesamiento de textos en español, comparando su desempeño bajo condiciones experimentales equivalentes. Esto permitiría determinar si otras arquitecturas pueden superar los resultados alcanzados por BERT en la tarea de clasificación multietiqueta desarrollada.

- Extender el alcance de la investigación mediante la incorporación de otros rasgos contemplados en el modelo de los Cinco Grandes Factores (*Big Five*), permitiendo analizar la viabilidad de aplicar la metodología desarrollada a dimensiones adicionales de personalidad. Esta ampliación requeriría conjuntos de datos específicamente etiquetados para cada rasgo y procesos de evaluación independientes.
- Continuar el desarrollo de la aplicación mediante la incorporación de mecanismos de almacenamiento persistente, generación de reportes y herramientas adicionales para el análisis de resultados. Estas funcionalidades podrían facilitar el uso de la propuesta en posteriores investigaciones, manteniendo claramente su finalidad académica y evitando interpretar las predicciones generadas como evaluaciones psicológicas o diagnósticos clínicos.

REFERENCIAS

[1] D. Jurafsky and J. H. Martin, “Book Review Speech and Language Processing (second edition)”.

[2] J. Hirschberg and C. D. Manning, “Advances in natural language processing,” *Science (1979).*, vol. 349, no. 6245, pp. 261–266, Jul. 2015, doi: 10.1126/SCIENCE.AAA8685; WEBSITE:WEBSITE:AAAS-SITE;JOURNAL:JOURNAL:SCIENCE;ISSUE:ISSUE:DOI.

[3] P. T. Costa and R. R. McCrae, “Neo Personality Inventory.,” *Encyclopedia of psychology, Vol. 5.*, pp. 407–409, Oct. 2004, doi: 10.1037/10520-172.

[4] O. P. John and S. Srivastava, “The Big-Five Trait Taxonomy: History, Measurement, and Theoretical Perspectives”.

[5] Y. R. Tausczik and J. W. Pennebaker, “The psychological meaning of words: LIWC and computerized text analysis methods,” *J. Lang. Soc. Psychol.*, vol. 29, no. 1, pp. 2

4–54, Mar. 2010, doi: 10.1177/0261927X09351676;WEBSITE:WEBSITE:SAGE;REQUESTEDJOURNAL:JOURNAL:JLSA;WGROUP:STRING:PUBLICATION.

[6] J. W. Pennebaker, R. L. Boyd, K. Jordan, and K. Blackburn, “The Development and Psychometric Properties of LIWC2015,” Sep. 15, 2015. Accessed: Apr. 05, 2026. [Online]. Available: http://hdl.handle.net/2152/31333

[7] C. C. Aggarwal, “Machine Learning for Text: An Introduction,” *Machine Learning for Text*, pp. 1–16, 2018, doi: 10.1007/978-3-319-73531-3_1.

[8] S. Hochreiter and J. Schmidhuber, “Long Short-Term Memory,” *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997, doi: 10.1162/NECO.1997.9.8.1735.

[9] I. Goodfellow, Y. Bengio, A. Courville, F. Bach, and N. Buduma, “Deep Learning,” *files.batistalab.com*, 2016, doi: 10.4258/HIR.

[10] A. Vaswani *et al.*, “Attention is All you Need,” *Adv. Neural Inf. Process. Syst.*, vol. 30, 2017.

[11] J. Devlin, M.-W. Chang, K. Lee, K. T. Google, and A. I. Language, “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding,” *Proceedings of the 2019 Conference of the North*, pp. 4171–4186, 2019, doi: 10.18653/V1/N19-1423.

[12] Y. Liu *et al.*, “RoBERTa: A Robustly Optimized BERT Pretraining Approach,” Jul. 2019, Accessed: Apr. 14, 2026. [Online]. Available: https://arxiv.org/pdf/1907.11692

[13] V. Sanh, L. Debut, J. Chaumond, and T. Wolf, “DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter,” Oct. 2019, Accessed: Apr. 14, 2026. [Online]. Available: https://arxiv.org/pdf/1910.01108

[14] N. Majumder, S. Poria, A. Gelbukh, and E. Cambria, “Deep Learning-Based Document Modeling for Personality Detection from Text,” *IEEE Intell. Syst.*, vol. 32, no. 2, pp. 74–79, Mar. 2017, doi: 10.1109/MIS.2017.23.

[15] L. Hu, H. He, D. Wang, Z. Zhao, Y. Shao, and L. Nie, “LLM vs Small Model? Large Language Model Based Text Augmentation Enhanced Personality Detection Model,” *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 38, no. 16, pp. 18234–18242, Mar. 2024, doi: 10.1609/AAAI.V38I16.29782.

[16] J. C. Garduño-Miralrio, D. Valle-Cruz, and J. L. Tapia-Fabela, “Comparación del reconocimiento multimodal de emociones en textos de Twitter,” *Programación matemática y software*, vol. 15, no. 1, pp. 9–16, Feb. 2023, doi: 10.30973/PROGMAT/2023.15.1/2.

[17] A. J. J. Alfaro, G. C. Barrera, N. K. V. Vázquez, J.-K. Ruiz-Garduño, and C.-T. González-Ramírez, “Análisis de emociones en textos en español mediante traducción Automática y modelos BERT Multilingües: Análisis de emociones en textos en español,” *RICT Revista de Investigación*

Científica, Tecnológica e Innovación, vol. 3, no. 6, pp. 1–7, Nov. 2025, doi: 10.5281/ZENODO.17525359.

[18] B. Thapa and G. Cofre, "Emotion Classification In-Context in Spanish," *Communications in Computer and Information Science*, vol. 2607 CCIS, pp. 141–155, 2025, doi: 10.1007/978-3-032-04657-4_10.

[19] T. B. Brown *et al.*, "Language Models are Few-Shot Learners," *Adv. Neural Inf. Process. Syst.*, vol. 33, pp. 1877–1901, 2020, Accessed: Apr. 05, 2026. [Online]. Available: https://commoncrawl.org/the-data/

[20] J. Snell, K. S. Twitter, and R. S. Zemel, "Prototypical Networks for Few-shot Learning".

[21] H. W. Chung *et al.*, "Scaling Instruction-Finetuned Language Models," *Journal of Machine Learning Research*, vol. 25, Oct. 2022, Accessed: Jul. 28, 2026. [Online]. Available: https://arxiv.org/pdf/2210.11416

[22] S. Longpre *et al.*, "The Flan Collection: Designing Data and Methods for Effective Instruction Tuning," *Proc. Mach. Learn. Res.*, vol. 202, pp. 22631–22648, Jan. 2023, Accessed: Jul. 28, 2026. [Online]. Available: https://arxiv.org/pdf/2301.13688

[23] J. M. Johnson and T. M. Khoshgoftaar, "Survey on deep learning with class imbalance," *Journal of Big Data 2019 6:1*, vol. 6, no. 1, pp. 27-, Mar. 2019, doi: 10.1186/S40537-019-0192-5.

[24] Z.-H. Zhou and M.-L. Zhang, "Multi-Label Learning".

[25] G. Tsoumakas, I. Katakis, and I. Vlahavas, "Mining Multi-label Data," *Data Mining and Knowledge Discovery Handbook*, pp. 667–685, 2009, doi: 10.1007/978-0-387-09823-4_34.

[26] S. Bird, E. Klein, and E. Loper, "Natural Language Processing with Python (Google eBook)," p. 504, 2009, Accessed: Apr. 15, 2026. [Online]. Available: http://books.google.com/books?id=KGIbfiiP1i4C&pgis=1

[27] "spaCy · Industrial-strength Natural Language Processing in Python." Accessed: Apr. 15, 2026. [Online]. Available: https://spacy.io/

[28] B. M. Lake, R. Salakhutdinov, and J. B. Tenenbaum, “Human-level concept learning through probabilistic program induction,” *Science (1979).*, vol. 350, no. 6266, pp. 1332–1338, Dec. 2015, doi: 10.1126/SCIENCE.AAB3050;PAGE:STRING:ARTICLE/CHAPTER.

[29] S. Godbole, A. Harpale, S. Sarawagi, and S. Chakrabarti, “Document Classification Through Interactive Supervision of Document and Term Labels,” *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 3202, pp. 185–196, 2004, doi: 10.1007/978-3-540-30116-5_19.

[30] M. Sokolova and G. Lapalme, “A systematic analysis of performance measures for classification tasks,” *Inf. Process. Manag.*, vol. 45, no. 4, pp. 427–437, Jul. 2009, doi: 10.1016/J.IPM.2009.03.002.

[31] “Plan Nacional de Desarrollo Ecuador No Se Detiene 2025 – 2029 – Secretaría Nacional de la Administración Pública y Planificación.” Accessed: Apr. 22, 2026. [Online]. Available: https://planificacion.presidencia.gob.ec/plan-nacional-de-desarrollo-2025-2029-ecuador-no-se-detiene/

[32] F. & Ponce *et al.*, “Artificial Intelligence A Modern Approach Fourth Edition,” 2021, Accessed: Apr. 14, 2026. [Online]. Available: https://lccn.loc.gov/2019047498

[33] B. Liu and C. Cardie, “Sentiment Analysis and Opinion Mining,” vol. 5, no. 1, p. 2012, 2014, doi: 10.1162/COLI.

[34] M. Mairesse, M. Walker, M. Mehl, and R. Moore, “Using Linguistic Cues for the Automatic Recognition of Personality in Conversation and Text,” *Journal of Artificial Intelligence Research*, vol. 30, pp. 457–500, 2007.

[35] R. J. . Larsen and D. M. . Buss, “Personality psychology : domains of knowledge about human nature,” 2021, Accessed: Jul. 29, 2026. [Online]. Available: https://books.google.com/books/about/Personality_Psychology.html?hl=es&id=JPeDzgEACAAJ

[36] B. B. Lahey, “Public Health Significance of Neuroticism,” *American Psychologist*, vol. 64, no. 4, pp. 241–256, May 2009, doi: 10.1037/A0015309.

[37] T. A. . Widiger, “The Oxford handbook of the five factor model,” p. 589, 2017.

[38] G. Matthews, I. J. Deary, and M. C. Whiteman, “Personality traits, Third edition,” *Personality Traits, Third Edition*, pp. 1–568, Jan. 2009, doi: 10.1017/CBO9780511812743.

[39] D. Jurafsky and J. H. Martin, “Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition,” *aclanthology.orgD Jurafsky, JH Martinaclanthology.org*, Accessed: Jul. 30, 2026. [Online]. Available: https://aclanthology.org/anthology-files/pdf/J/J00/J00-4006.pdf

[40] W. J. M. Levelt, “Speaking: From intention to articulation. ACL-MIT Press series in natural-language processing.,” p. 565, 1989.

[41] T. A. Harley, “The Psychology of Language : From Data to Theory,” Dec. 2013, doi: 10.4324/9781315782942.

[42] M. Marczak and I. Coyne, “Cyberpsychology Edited by Alison Attrill,” pp. 145–163, 2015.

[43] A. J.-R. iberoamericana de educación a distancia and undefined 2003, “Understanding the psychology of Internet behaviour: Virtual worlds, real lives,” *researchgate.netAN JoinsonRevista iberoamericana de educación a distancia, 2003•researchgate.net*, Accessed: Jul. 29, 2026. [Online]. Available: https://www.researchgate.net/profile/Adam-Joinson/publication/42788680_Understanding_the_Psychology_of_Internet_Behavior_Virtual_Worlds_Real_Lives/links/00b7d53391f5d467e9000000/Understanding-the-Psychology-of-Internet-Behavior-Virtual-Worlds-Real-Lives.pdf

[44] A. Barak and J. Suler, “Reflections on the Psychology and Social Science of Cyberspace”, Accessed: Jul. 30, 2026. [Online]. Available: http://www.pewinternet.org

[45] T. Young, D. Hazarika, S. Poria, and E. Cambria, “Recent trends in deep learning based natural language processing [Review Article],” *IEEE Comput. Intell. Mag.*, vol. 13, no. 3, pp. 55–75, Aug. 2018, doi: 10.1109/MCI.2018.2840738.

[46] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. 2016. Accessed: Apr. 14, 2026. [Online]. Available: https://books.google.com/books?hl=en&lr=&id=omivDQAAQBAJ&oi=fnd&pg=PR5&ots=MOT4dmkAOU&sig=kFKbQMeU48tOW7tMMnjuSnnpP2A

[47] Y. Lecun, Y. Bengio, and G. Hinton, “Deep learning,” *Nature*, vol. 521, no. 7553, pp. 436–444, May 2015, doi: 10.1038/NATURE14539;SUBJMETA.

[48] T. Kudo and J. Richardson, “SentencePiece: A simple and language independent subword tokenizer and detokenizer for Neural Text Processing,” *EMNLP 2018 - Conference on Empirical Methods in Natural Language Processing: System Demonstrations, Proceedings*, pp. 66–71, 2018, doi: 10.18653/V1/D18-2012.

[49] C. Manning and H. Schutze, “Foundations of statistical natural language processing,” 1999, Accessed: Jul. 30, 2026. [Online]. Available: https://books.google.com.ec/books?hl=es&lr=&id=ZL34DwAAQBAJ&oi=fnd&pg=PR16&dq=Manning,+C.+D.,+%26+Sch%C3%BCtze,+H.+(1999).+Foundations+of+Statistical+Natural+Language+Processing.+MIT+Press.&ots=fm0YbgsNTC&sig=iFn1gbN5HYiOBLYcmVr-UO1bJhg

[50] Y. Goldberg, Y. Liu, and M. Zhang, “Neural Network Methods for Natural Language Processing,” vol. 37, 2017, doi: 10.2200/S00762ED1V01Y201703HLT037.

[51] T. Mikolov, K. Chen, G. Corrado, and J. Dean, “Efficient Estimation of Word Representations in Vector Space”, Accessed: Jul. 30, 2026. [Online]. Available: http://ronan.collobert.com/senna/

[52] J. Pennington, R. Socher, and C. D. Manning, “GloVe: Global Vectors for Word Representation,” pp. 1532–1543, Accessed: Jul. 30, 2026. [Online]. Available: http://nlp.

[53] M. E. Peters *et al.*, “Deep Contextualized Word Representations,” *NAACL HLT 2018 - 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies - Proceedings of the Conference*, vol. 1, pp. 2227–2237, 2018, doi: 10.18653/V1/N18-1202.

[54] X. Han *et al.*, “Pre-trained models: Past, present and future,” *AI Open*, vol. 2, pp. 225–250, Jan. 2021, doi: 10.1016/J.AIOPEN.2021.08.002.

[55] X. P. Qiu, T. X. Sun, Y. G. Xu, Y. F. Shao, N. Dai, and X. J. Huang, “Pre-trained models for natural language processing: A survey,” *SpringerX Qiu, T Sun, Y Xu, Y Shao, N Dai, X HuangScience China technological sciences, 2020•Springer*, vol. 63, no. 10, pp. 1872–1897, Oct. 2020, doi: 10.1007/S11431-020-1647-3.

[56] T. Lin, Y. Wang, X. Liu, and X. Qiu, “A survey of transformers,” *AI Open*, vol. 3, pp. 111–132, Jan. 2022, doi: 10.1016/J.AIOPEN.2022.10.001.

[57] R. Bommasani *et al.*, “On the Opportunities and Risks of Foundation Models,” Aug. 2021, Accessed: Jul. 30, 2026. [Online]. Available: https://arxiv.org/pdf/2108.07258

[58] A. Rogers, O. Kovaleva, and A. Rumshisky, “A Primer in BERTology: What We Know About How BERT Works,” *Trans. Assoc. Comput. Linguist.*, vol. 8, pp. 842–866, 2020, doi: 10.1162/TACL_A_00349.

[59] C. Raffel *et al.*, “Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer,” *Journal of Machine Learning Research*, vol. 21, no. 140, pp. 1–67, 2020, Accessed: Jul. 30, 2026. [Online]. Available: http://jmlr.org/papers/v21/20-074.html

[60] J. Wei *et al.*, “Finetuned Language Models Are Zero-Shot Learners,” *ICLR 2022 - 10th International Conference on Learning Representations*, Sep. 2021, Accessed: Jul. 30, 2026. [Online]. Available: https://arxiv.org/pdf/2109.01652

[61] M. L. Zhang and Z. H. Zhou, “A review on multi-label learning algorithms,” *IEEE Trans. Knowl. Data Eng.*, vol. 26, no. 8, pp. 1819–1837, 2014, doi: 10.1109/TKDE.2013.39.

[62] M. S.-O. S. University, undefined Corvallis, and undefined 2010, “A literature survey on algorithms for multi-label learning,” *researchgate.netMS SorowerOregon State University, Corvallis, 2010•researchgate.net*, Accessed: Jul. 30, 2026. [Online]. Available: https://www.researchgate.net/profile/Mohammad-Sorower-2/publication/266888594_A_Literature_Survey_on_Algorithms_for_Multi-label_Learning/links/57fdb69308ae6750f80665b5/A-Literature-Survey-on-Algorithms-for-Multi-label-Learning.pdf

[63] Y. Wang, J. T. Kwok, L. M. Ni, and H. Kong, “Generalizing from a Few Examples: A Survey on Few-Shot Learning,” 2020, doi: 10.1145/3386252.

[64] T. B. Brown *et al.*, “Language Models are Few-Shot Learners,” *Adv. Neural Inf. Process. Syst.*, vol. 33, pp. 1877–1901, 2020, Accessed: Jul. 30, 2026. [Online]. Available: https://commoncrawl.org/the-data/

[65] T. B. Brown *et al.*, “Language Models are Few-Shot Learners,” *Adv. Neural Inf. Process. Syst.*, vol. 33, pp. 1877–1901, 2020, Accessed: Apr. 05, 2026. [Online]. Available: https://commoncrawl.org/the-data/

[66] D. M. W. Powers and Ailab, “Evaluation: from precision, recall and F-measure to ROC, informedness, markedness and correlation,” Oct. 2020, Accessed: Jul. 30, 2026. [Online]. Available: https://arxiv.org/pdf/2010.16061

[67] A. Géron, “Hands-on machine learning with Scikit-Learn, Keras and TensorFlow : concepts, tools, and techniques to build intelligent systems,” p. 834, 2023.

[68] “3.14.6 Documentation.” Accessed: Jul. 30, 2026. [Online]. Available: https://docs.python.org/3/

[69] W. McKinney, “Python for data analysis: Data wrangling with Pandas, NumPy, and IPython,” 2012, Accessed: Jul. 30, 2026. [Online]. Available: https://books.google.com.ec/books?hl=es&lr=&id=v3n4_AK8vu0C&oi=fnd&pg=PR3&dq=McKinney,+W.++Python+for+Data+Analysis.&ots=riLJ7nzuoD&sig=Fe7LqcMQADlVZBKKRq285Sz8mL0

[70] E. Bisong, “Google Colaboratory,” *Building Machine Learning and Deep Learning Models on Google Cloud Platform*, pp. 59–64, 2019, doi: 10.1007/978-1-4842-4470-8_7.

[71] “Google Colab.” Accessed: Jul. 30, 2026. [Online]. Available: https://colab.research.google.com/

[72] E. Bisong, “Building machine learning and deep learning models on Google cloud platform,” 2019, Accessed: Jul. 30, 2026. [Online]. Available: https://link.springer.com/content/pdf/10.1007/978-1-4842-4470-8.pdf

[73] G. Van Rossum and F. L. Drake, “Python/C API Manual - Python 3,” p. 294, 2009.

[74] “The Python Language Reference Guido van Rossum and the Python development team,” 2021.

[75] W. Mckinney, “pandas: a Foundational Python Library for Data Analysis and Statistics”, Accessed: Apr. 15, 2026. [Online]. Available: http://pandas.sf.net

[76] C. R. Harris *et al.*, “Array programming with NumPy,” *Nature 2020 585:7825*, vol. 585, no. 7825, pp. 357–362, Sep. 2020, doi: 10.1038/s41586-020-2649-2.

[77] S. Bird, “NLTK Documentation Release 3.2.5,” 2017.

[78] “spaCy · Industrial-strength Natural Language Processing in Python.” Accessed: Jul. 30, 2026. [Online]. Available: https://spacy.io/

[79] M. H.-(No Title) and undefined 2017, “spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing,” *cir.nii.ac.jp*, Accessed: Jul. 30, 2026. [Online]. Available: https://cir.nii.ac.jp/crid/1370021390573874949

[80] T. Wolf *et al.*, “Transformers: State-of-the-Art Natural Language Processing”, Accessed: Apr. 15, 2026. [Online]. Available: https://github.com/huggingface/

[81] “Transformers · Hugging Face.” Accessed: Jul. 30, 2026. [Online]. Available: https://huggingface.co/docs/transformers/index

[82] J. Hunter and D. Dale, “The Matplotlib User’s Guide,” 2007.

[83] “Matplotlib documentation — Matplotlib 3.11.1 documentation.” Accessed: Jul. 30, 2026. [Online]. Available: https://matplotlib.org/stable/

[84] M. L. Waskom, “seaborn: statistical data visualization”, doi: 10.21105/joss.03021.

[85] M. Waskom, “seaborn: statistical data visualization,” *J. Open Source Softw.*, vol. 6, no. 60, p. 3021, Apr. 2021, doi: 10.21105/JOSS.03021.

[86] J. W. Pennebaker, “The secret life of pronouns,” *New Sci. (1956).*, vol. 211, no. 2828, pp. 42–45, Sep. 2011, doi: 10.1016/S0262-4079(11)62167-2.

[87] R. L. Boyd and J. W. Pennebaker, “Language-based personality: a new approach to personality in a digital world,” *Curr. Opin. Behav. Sci.*, vol. 18, pp. 63–68, Dec. 2017, doi: 10.1016/J.COBEHA.2017.07.017.

[88] J. W. Pennebaker, R. J. Booth, R. L. Boyd, and M. E. Francis, “Linguistic Inquiry and Word Count: LIWC2015 Operator’s Manual”, Accessed: Apr. 22, 2026. [Online]. Available: www.LIWC.net

[89] F. Mairesse, M. A. Uk, M. R. Mehl, and R. K. Moore, “Using Linguistic Cues for the Automatic Recognition of Personality in Conversation and Text,” *Journal of Artificial Intelligence Research*, vol. 1.

[90] W. Youyou, M. Kosinski, and D. Stillwell, “Computer-based personality judgments are more accurate than those made by humans,” *Proc. Natl.*

Acad. Sci. U. S. A., vol. 112, no. 4, pp. 1036–1040, Jan. 2015, doi: 10.1073/PNAS.1418680112;WGROUP:STRING:PUBLICATION.

[91] E. M. Bender and A. Koller, “Climbing towards NLU: On Meaning, Form, and Understanding in the Age of Data,” pp. 5185–5198, Accessed: Apr. 15, 2026. [Online]. Available: https://www.nytimes.com/2018/11/18/technology/artific

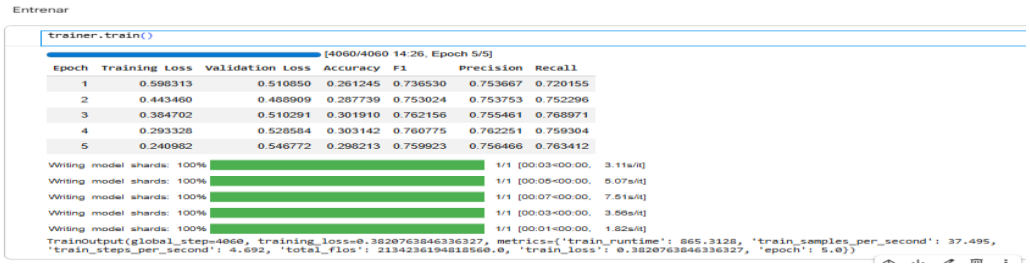
ANEXOS

Anexo 1. Estructura del conjunto de datos utilizado en la investigación

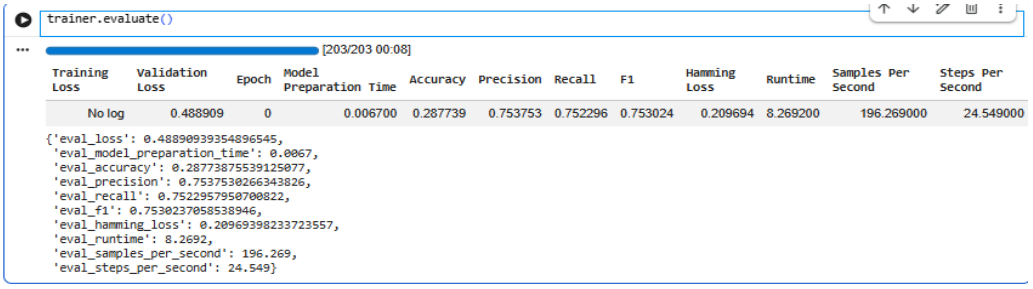
No.	TWEETS	VERGÜENZA	IRRITABILIDA	INSEGURIDA	DEPRESION	IRA	ANSIEDAD
1	No puedo creer la vergüenza que siento por hablar mal de ti frente a otros.	1	0	1	0	0	0
2	Lamentablemente he pasado vergüenza al hablar mal de ti durante la cena.	1	1	0	0	1	1
3	Es humillante mentir sobre mi ubicación de forma impulsiva, perdóname por favor.	1	1	0	0	0	0
4	Me da vergüenza haber impedir tus planes hace un momento.	1	0	0	0	1	1
5	Qué vergüenza descuidar tu petición en la reunión, lo lamento de corazón.	1	0	0	0	1	0
6	No puedo creer la vergüenza que siento por hacerte esperar sin aviso durante la cena.	1	1	0	0	1	0
7	Me da vergüenza haber no apoyarte cuando lo necesitabas de forma impulsiva.	1	0	0	0	1	0
8	Es humillante mentir sobre mi ubicación el otro día, perdóname por favor.	1	0	1	0	1	0
9	Qué vergüenza hablar mal de ti en la reunión, lo lamento de corazón.	1	1	1	0	1	0
10	Me avergüenza haber descuidar tu petición de forma impulsiva.	1	0	1	1	0	0
11	Siento tanta vergüenza por haber hablar mal de ti ayer.	1	0	0	1	0	0
12	Ha sido vergonzoso hablar mal de ti sin consultarte, lo reconozco.	1	1	1	1	0	0
13	No puedo creer la vergüenza que siento por ser indiferente a tus sentimientos hace un r	1	0	0	0	1	0
14	Ha sido vergonzoso culparte frente a otros de forma impulsiva, lo reconozco.	1	0	1	1	0	0
15	Lamentablemente he pasado vergüenza al descuidar tu petición en público.	1	0	1	0	1	0
16	Reconozco la vergüenza de ignorar tus consejos ayer, lo siento.	1	1	0	0	1	1
17	Reconozco la vergüenza de impedir tus planes sin medir las consecuencias, lo siento.	1	0	0	1	1	0

Anexo 1. Muestra de la estructura del conjunto de datos utilizado en la investigación. Elaboración propia.

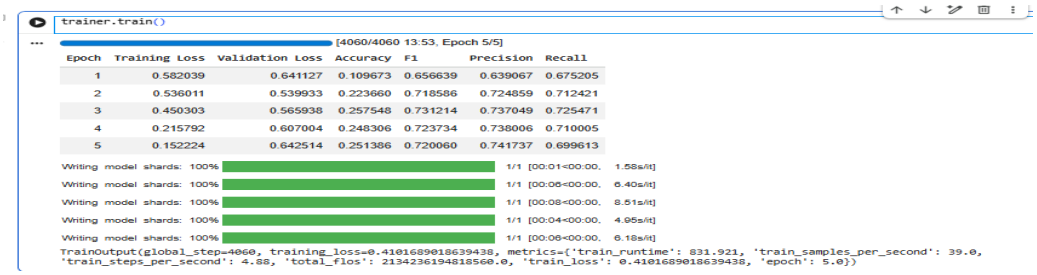
Anexo 2. Evidencias del entrenamiento y evaluación de los modelos Transformer



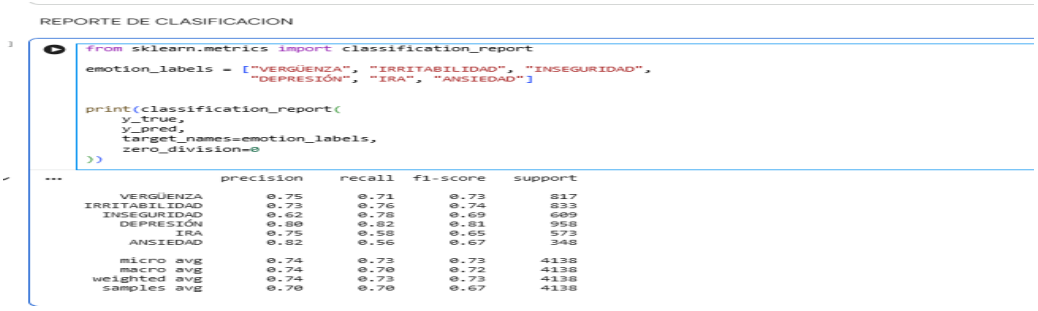
Anexo 2. Entrenamiento del modelo BERT durante cinco épocas en Google Colab. Elaboración propia.



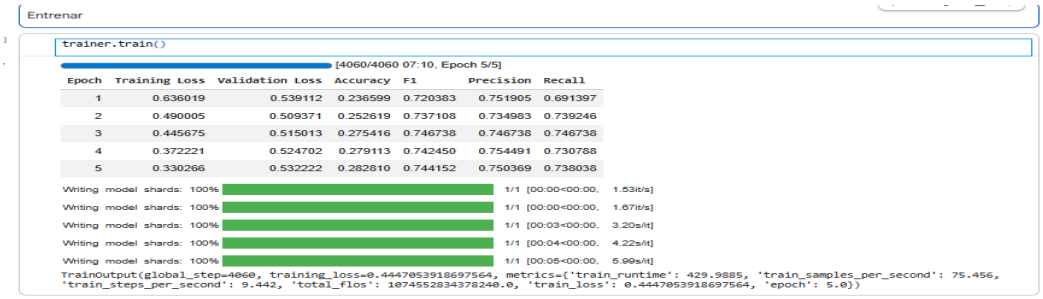
Anexo 3. Evaluación del modelo BERT mediante las métricas definidas para la clasificación multietiqueta. Elaboración propia.



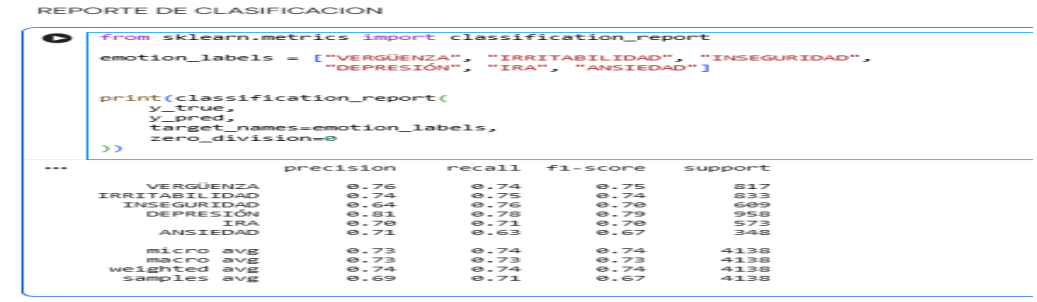
Anexo 4. Entrenamiento del modelo RoBERTa durante cinco épocas en Google Colab. Elaboración propia.



Anexo 5. Reporte de clasificación por indicador lingüístico obtenido mediante el modelo RoBERTa. Elaboración propia

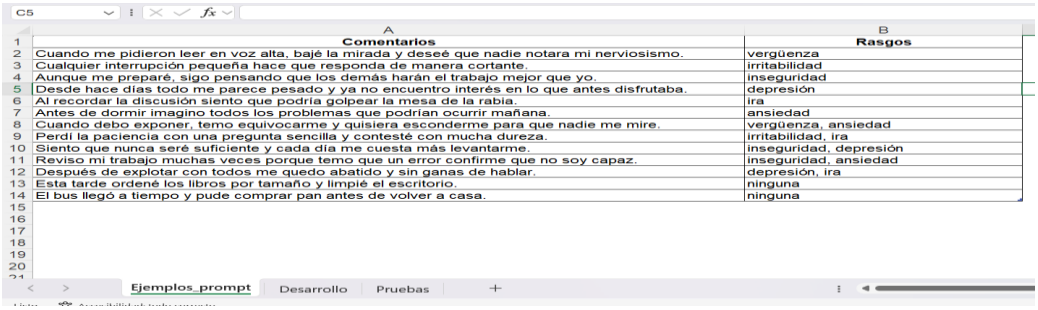


Anexo 6. Entrenamiento del modelo DistilBERT durante cinco épocas en Google Colab. Elaboración propia.



Anexo 7. Reporte de clasificación por indicador lingüístico obtenido mediante el modelo DistilBERT. Elaboración propia.

Anexo 3. Implementación de flan-t5 mediante Few-Shot Learning



Anexo 8. Organización de los conjuntos utilizados para la implementación de FLAN-T5 mediante Few-Shot Learning. Elaboración propia.

8. Demostraciones Few-Shot por etiqueta

```
MAX_DEMOS_POSITIVAS=2
MAX_DEMOS_NEGATIVAS=2

def seleccionar_demostraciones(etiqueta):
    pos=df_prompt[df_prompt['rasgos_lista'].apply(lambda x: etiqueta in x)].head(MAX_DEMOS_POSITIVAS)
    neg=df_prompt[df_prompt['rasgos_lista'].apply(lambda x: etiqueta not in x)].head(MAX_DEMOS_NEGATIVAS)
    demos=pd.concat([pos,neg],ignore_index=True)
    return demos.sample(frac=1,random_state=SEED).reset_index(drop=True) if len(demos)>1 else demos

for e in ETIQUETAS:
    print('\n',e.upper())
    display(seleccionar_demostraciones(e)[['Comentarios','Rasgos']])
```

VERGÜENZA

	Comentarios	Rasgos
0	Cuando debo exponer, temo equivocarme y quisie...	vergüenza, ansiedad
1	Aunque me preparé, sigo pensando que los demás...	inseguridad
2	Cuando me pidieron leer en voz alta, bajé la m...	vergüenza
3	Cualquier interrupción pequeña hace que respon...	irritabilidad

IRRITABILIDAD

	Comentarios	Rasgos
0	Perdí la paciencia con una pregunta sencilla y...	irritabilidad, ira
1	Aunque me preparé, sigo pensando que los demás...	inseguridad
2	Cualquier interrupción pequeña hace que respon...	irritabilidad
3	Cuando me pidieron leer en voz alta, bajé la m...	vergüenza

INSEGURIDAD

	Comentarios	Rasgos
0	Siento que nunca seré suficiente y cada día me...	inseguridad, depresión
1	Cualquier interrupción pequeña hace que respon...	irritabilidad
2	Aunque me preparé, sigo pensando que los demás...	inseguridad
3	Cuando me pidieron leer en voz alta, bajé la m...	vergüenza

Anexo 9. Selección de demostraciones Few-Shot utilizadas para la clasificación de los indicadores lingüísticos. Elaboración propia.

12. Ajustar umbrales por etiqueta con Desarrollo

```
def vector_real(df,etiqueta):
    return np.array([int(etiqueta in r) for r in df['rasgos_lista']])

def buscar_umbral_optimo(y_real,probs):
    mejor=(0.50,-1)
    for u in np.arange(0.10,0.91,0.05):
        f1=f1_score(y_real,(probs>=u).astype(int),zero_division=0)
        if f1>mejor[1]: mejor=(float(u),f1)
    return mejor

UMBRALES={}
res=[]
for e in ETIQUETAS:
    y=vector_real(resultados_desarrollo,e)
    probs=resultados_desarrollo[f'prob_{e}'].to_numpy(float)
    u,f1=buscar_umbral_optimo(y,probs)
    UMBRALES[e]=u
    res.append({'Etiqueta':e,'Umbral':u,'F1 Desarrollo':f1,'Positivos reales':int(y.sum())})
df_umbrales=pd.DataFrame(res)
display(df_umbrales.style.format({'Umbral':'{:.2f}','F1 Desarrollo':'{:.4f}'}))
print('UMBRALES:',UMBRALES)
```

	Etiqueta	Umbral	F1 Desarrollo	Positivos reales
0	vergüenza	0.20	0.5000	6
1	irritabilidad	0.20	0.4000	4
2	inseguridad	0.15	0.3030	7
3	depresión	0.20	0.4286	8
4	ira	0.10	0.2000	5
5	ansiedad	0.15	0.3226	8

UMBRALES: {'vergüenza': 0.20000000000000004, 'irritabilidad': 0.20000000000000004, 'inseguridad': 0.15000000000000002, 'depresión': 0.200000

Anexo 10. Ajuste de los umbrales de decisión por indicador mediante el conjunto de Desarrollo. Elaboración propia.

```
print('F1 macro presencia/ausencia:',f'{mejor_global["F1 macro"]:.4f}')
display(df_umbral_global.sort_values(['Balanced Accuracy','F1 macro'],ascending=False).head(10).style.format({'
```

```
Comentarios con algún rasgo: 30
Comentarios sin rasgos: 20

MEJOR UMBRAL GLOBAL
Umbral global: 0.19
Balanced Accuracy: 0.7250
F1 macro presencia/ausencia: 0.7159
```

	Umbral	Balanced Accuracy	F1 macro	Precision presencia	Recall presencia
9	0.19	0.7250	0.7159	0.8077	0.7000
10	0.20	0.7083	0.6970	0.8000	0.6667
8	0.18	0.6667	0.6667	0.7333	0.7333
12	0.22	0.6583	0.6186	0.8235	0.4667
11	0.21	0.6500	0.6198	0.7895	0.5000
13	0.23	0.6500	0.5942	0.8571	0.4000
7	0.17	0.6250	0.6264	0.6857	0.8000
14	0.24	0.6000	0.5246	0.8182	0.3000
6	0.16	0.5917	0.5864	0.6579	0.8333
5	0.15	0.5667	0.5559	0.6410	0.8333

Anexo 11. Selección del umbral global de abstención mediante el conjunto de Desarrollo.

Nº	Comentarios	Rasgos	Prediccion
1	No te humillo, solo te muestro lo que eres.	vergüenza, irritabilidad	ninguna
2	Te voy a borrar esa sonrisa de la cara.	irritabilidad, inseguridad, ira, ansiedad	ninguna
3	Merida tierra de Anarquía los "moteros" de nuevo haciendo de las suyas en el Municipio Libertador... Otro año más el merideño es marginado y pisados sus derechos por estos "tur...	ninguna	ninguna
4	Es increíble que hasta con mis amigas me termino quedando sola. Ya no soporto que me ignoren.	vergüenza, irritabilidad, depresión, ira, ansiedad	inseguridad, depresión, ira, ansiedad
5	Te haré papilla.	irritabilidad, ira	ninguna
6	Se fumaron la luna de miel de 100 días en 20 días de gobierno. Desconocen espiritualmente al pueblo argentino, lo van a hacer llegar agotado e irascible al inicio del ciclo lec...	irritabilidad, ira	ninguna
7	Me encerré en el baño durante horas solo para evitar enfrentar lo que pasaba afuera.	vergüenza, irritabilidad, inseguridad, depresión, ira, ansiedad	inseguridad, depresión, ira, ansiedad
8	ya weon q me diga q me kiere mandar a la chucha si anda ma distante konmigo	ninguna	ninguna
9	Tengo mucho trabajo, por eso no voy a tu salida, pero en realidad quiero descansar	vergüenza, irritabilidad	ninguna
10	Es que estaba claro que en las fiestas DENI y BAUTI se evitan todo el rato entre lo timido que	depresión	ninguna

Anexo 12. Evaluación final de FLAN-T5 sobre 120 comentarios independientes del conjunto de Pruebas. Elaboración propia.

Anexo 4. Interfaz de la aplicación desarrollada

Anexo 13. Interfaz principal de la aplicación desarrollada para la detección de indicadores lingüísticos asociados al neuroticismo. Elaboración propia.