UNIVERSIDAD EUROPEA DE MADRID

ESCUELA DE ARQUITECTURA, INGENIERÍA Y DISEÑO

MÁSTER UNIVERSITARIO EN ANALISIS DE DATOS MASIVOS (BIG DATA)

TRABAJO FIN DE MÁSTER

BIG DATA Y ANÁLISIS DE SENTIMIENTO: ESTUDIO DE LA OPINIÓN GLOBAL SOBRE NOTICIAS EN TIEMPO REAL

JOSE RICARDO NIETO PRIMERA

Dirigido por

SAMUEL GARCÍA SABOYA

CURSO 2024-2025

TÍTULO: BIG DATA Y ANÁLISIS DE SENTIMIENTO: ESTUDIO DE LA OPINIÓN GLOBAL SOBRE NOTICIAS EN TIEMPO REAL

AUTOR: JOSE RICARDO NIETO PRIMERA

TITULACIÓN: MÁSTER UNIVERSITARIO EN ANÁLISIS DE DATOS MASIVOS (BIG DATA)

DIRECTOR DEL PROYECTO: SAMUEL GARCÍA SABOYA

FECHA: JULIO DE 2025

RESUMEN

En la actualidad, la conformación de la opinión pública sobre noticias ocurre de forma inmediata y global, principalmente a través de titulares difundidos en plataformas digitales. Sin embargo, los sistemas de análisis de sentimiento suelen centrarse en redes sociales y textos en inglés, presentando limitaciones en multilingüismo, tiempo real y granularidad geográfica.

En este trabajo, se desarrolla una solución integral basada en Big Data y Procesamiento de Lenguaje Natural (NLP). El sistema es capaz de recopilar, limpiar, analizar y visualizar en tiempo real el sentimiento de titulares de noticias multilingües (inglés y español), con enriquecimiento geográfico.

Su arquitectura modular de microservicios (Docker) automatiza la ingesta desde cuatro APIs, aplica limpieza avanzada y evalúa los titulares con un enfoque hibrido de modelos, combinando enfoques léxicos y basados en Transformer como XLM-RoBERTa y RoBERTuito.

Validado sobre 120.000 titulares etiquetados manualmente, el estudio no solo confirma el alto rendimiento de XLM-RoBERTa, sino que revela dos hallazgos científicos clave: la “Paradoja del Preprocesamiento”, que demuestra el impacto marginal de la limpieza de textos en modelos Transformer, y el “Problema de la Neutralidad”, una incapacidad sistemática de los modelos actuales para clasificar titulares neutros.

El proyecto aporta una solución técnica reproducible y, fundamentalmente, evidencia empírica que cuestiona los paradigmas de clasificación de sentimiento en el periodismo. Los resultados validan la viabilidad del sistema y abren nuevas vías de investigación sobre los límites del análisis de sentimiento en lenguaje mediático.

Palabras clave: Big Data, análisis de sentimiento, NLP, titulares de noticias, Problema de la Neutralidad, Paradoja del Preprocesamiento, arquitectura de microservicios, geolocalización, modelos Transformer

ABSTRACT

Nowadays, public opinion on news is shaped instantly and globally, mainly through headlines disseminated on digital platforms. However, most sentiment analysis systems focus on social media and English texts, facing limitations in multilingualism, real-time processing, and geographic granularity.

This master's thesis develops an integrated solution based on Big Data and Natural Language Processing (NLP) capable of collecting, cleaning, analyzing, and visualizing in real time the sentiment of multilingual news headlines (English and Spanish), enriched with geographic context.

The system implements a modular microservices architecture (Docker) that automates ingestion from four news APIs, applies advanced cleaning, and evaluates headlines with a hybrid model approach combining lexical techniques and Transformer-based models such as XLM-RoBERTa and RoBERTuito.

Validated on 120,000 manually labeled headlines, the study not only confirms the high performance of XLM-RoBERTa but also unveils two key scientific findings: the "Preprocessing Paradox", showing the marginal impact of text cleaning on Transformer models, and the "Neutrality Problem", highlighting the systematic inability of current models to classify neutral headlines.

The project contributes a reproducible technical solution and, fundamentally, empirical evidence that challenges prevailing paradigms of sentiment classification in journalism. The results confirm the feasibility of the system and open new avenues for research on the boundaries of sentiment analysis in media language.

Keywords: Big Data, sentiment analysis, NLP, news headlines, Neutrality Problem, Preprocessing Paradox, microservices architecture, geolocation, Transformer models

DEDICATORIA

A Dios, por hablar desde el silencio de lo cotidiano.

A mi madre, pro enseñarme el abecedario antes de los tres años, compartiendo su amor por la comunicación.

A mi padre, por revelarme el valor de las letras en la historia, y con ellas, su pasión y pensamiento crítico.

A mi esposa, por ser mi más fiel emisor y receptor de sentimientos.

TABLA RESUMEN

	DATOS
Nombre y apellidos:	Jose Ricardo Nieto Primera
Título del proyecto:	Big Data y Análisis de Sentimiento: Estudio de la Opinión Global sobre Noticias en Tiempo Real
Director del proyecto:	Samuel García Saboya
El proyecto se ha realizado en colaboración de una empresa o a petición de una empresa:	NO
El proyecto ha implementado un producto:	SI
El proyecto ha consistido en el desarrollo de una investigación o innovación:	SI
Objetivo general del proyecto:	Desarrollar un sistema modular y escalable basado en tecnologías Big Data y Procesamiento de Lenguaje Natural (NLP) para el análisis en tiempo real del sentimiento asociado a titulares de noticias de múltiples fuentes digitales, con capacidad de enriquecimiento geográfico y visualización interactiva estructurada a nivel geográfico y temporal.

Índice

Índice de Figuras

Índice de Tablas

Capítulo 1. RESUMEN DEL PROYECTO

1.1 Contexto y justificación

El consumo de información ha experimentado una transformación profunda en la era digital. Las noticias, tradicionalmente transmitidas a través de medios impresos y audiovisuales, hoy circulan en tiempo real mediante plataformas digitales y redes sociales. En este nuevo ecosistema informativo, los titulares adquieren el papel protagónico, pues funcionan como vehículos inmediatos de impacto emocional y construcción de opinión pública. En consecuencia, analizar el sentimiento que transmiten estos titulares se ha convertido en una necesidad emergente para comprender dinámicas sociales, tendencias mediáticas y polarización discursiva.

A pesar de los avances en procesamiento de lenguaje natural (NLP) y sistemas de análisis de sentimiento, persisten limitaciones relevantes: la mayoría de las soluciones están orientadas a redes sociales, centradas en idioma inglés, con escasa granularidad geográfica y sin capacidad de operar en tiempo real. Este proyecto aborda esta brecha mediante una solución técnica basada en tecnologías Big Data y modelos NLP multilingües, orientada al análisis de sentimiento de titulares de noticias globales y, a su vez, explora interrogantes fundamentales sobre la aplicabilidad de las técnicas estándar de NLP en el complejo dominio periodístico.

1.2 Planteamiento del problema

La carencia de sistemas capaces de integrar múltiples fuentes de noticias, procesar titulares en distintos idiomas y ofrecer análisis emocional en tiempo real con enriquecimiento geográfico impide una evaluación precisa y continua del estado de opinión pública global. Esto limita la capacidad de instituciones, medios y ciudadanos para interpretar el tono emocional dominante en la cobertura mediática.

Más allá de esta carencia técnica, el problema se extiende a un nivel conceptual. ¿Son las categorías de sentimiento tradicionales (positivo, neutro, negativo) adecuadas para el lenguaje periodístico, inherentemente diseñado para ser persuasivo y ambiguo? ¿Qué impacto real tienen los complejos pipelines de preprocesamiento de texto en el rendimiento de los modernos y robustos modelos Transformer?

En este contexto, se propone el desarrollo de una solución integral que aborde de forma simultánea los desafíos de multilingüismo, tiempo real y visualización geolocalizada, al tiempo que aporta evidencia empírica para responder estas preguntas críticas en el campo del NLP.

1.3 Objetivos del proyecto

El objetivo general consiste en diseñar e implementar un sistema modular y escalable para el análisis de sentimiento de titulares de noticias globales, en tiempo real, empleando arquitecturas Big Data y modelos avanzados de procesamiento de lenguaje natural.

Como objetivos específicos se destacan:

- la ingesta automatizada desde múltiples APIs de noticias,
- la limpieza lingüística de los titulares,
- la evaluación mediante modelos híbridos (léxicos y basados en Transformers),
- la validación mediante anotación humana,
- y la visualización interactiva de los resultados en un dashboard multilingüe y georreferenciado.

1.4 Resultados obtenidos

El sistema desarrollado recopila, procesa y analiza más de 3.000 titulares diarios en inglés y español, con evaluación de sentimiento mediante cinco modelos: VADER, DistilBERT, XLM-RoBERTa, FinBERT y RoBERTuito. El mejor rendimiento se observó en XLM-RoBERTa, con una precisión del 62,1% y un coeficiente Kappa de 0,429 frente a una muestra de 120.000 titulares etiquetados manualmente.

De forma crucial, la investigación reveló dos hallazgos científicos que trascienden las métricas de rendimiento: la “Paradoja del Preprocesamiento”, que demuestra el impacto marginal de la limpieza de texto en los modelos Transformer, y el “Problema de la Neutralidad”, una incapacidad sistemática de todos los modelos para clasificar correctamente los titulares neutros. Estos descubrimientos constituyen una de las principales aportaciones de la tesis.

El sistema opera sobre una arquitectura distribuida basada en contenedores Docker y MongoDB, con visualización en Streamlit. Se alcanzó una alta eficiencia en consulta gracias a optimizaciones con índices, vistas especializadas y caché inteligente.

1.5 Estructura de la memoria

La presente memoria se organiza en nueve capítulos. El **Capítulo 2** revisa el estado del arte y los fundamentos teóricos del análisis de sentimiento en titulares de noticias. El **Capítulo 3** expone los objetivos del proyecto y sus beneficios. El **Capítulo 4** describe la solución técnica, la planificación, los recursos, presupuesto y resultados obtenidos. El **Capítulo 5** presenta una discusión crítica de los hallazgos. El **Capítulo 6** recoge las conclusiones, mientras que el **Capítulo 7** sugiere futuras líneas de trabajo. Finalmente, los **Capítulos 8** y **9** recogen las referencias bibliográficas y los anexos técnicos respectivamente.

Capítulo 2. ANTECEDENTES / ESTADO DEL ARTE

2.1 Evolución del consumo digital en noticias y formación de opinión

La manera en que se consume la información ha evolucionado notablemente a lo largo del tiempo. Con la invención de la imprenta, el acceso a las noticias se expandió, aunque seguía siendo un privilegio limitado a ciertos sectores sociales. Durante siglos, el formato impreso fue el canal dominante de difusión informativa. Sin embargo, la globalización y la llegada de internet transformaron profundamente este escenario.

Según Reuters Institute Digital News Report 2023, aproximadamente el 72% de los usuarios de internet en Europa acceden a noticias a través de plataformas digitales, mientras que, en Estados Unidos, el 48% de los adultos obtienen noticias frecuentemente desde redes sociales [1]. Actualmente, la mayoría de las personas no accede a la información mediante periódicos en papel, sino a través de sus dispositivos móviles, donde los titulares son presentados en forma de notificaciones, banners o contenidos destacados en plataformas digitales.

En este nuevo paradigma, el titular adquiere una importancia crítica. La mayoría de los usuarios decide si una noticia merece su atención basándose únicamente en el impacto del título. Muchas veces, no se accede al contenido completo del artículo, lo que convierte al titular en la principal fuente de información y, en consecuencia, de formación de opinión. Este fenómeno amplifica el poder emocional del lenguaje y abre la puerta al análisis de sentimiento sobre textos breves, cargados de intencionalidad y diseño persuasivo.

Figura 2.1: Evolución del consumo de noticias

2.2 Estado del arte en análisis de sentimiento de titulares de noticias

2.2.1 Enfoques tradicionales y limitaciones

El análisis de sentimiento se ha aplicado históricamente a plataformas como redes sociales, foros o sistemas de atención al cliente, donde predominan textos breves y espontáneos. Entre los modelos más utilizados destacan VADER y TextBlob, por su simplicidad y eficacia para detectar emociones en frases cortas, aunque presentan limitaciones en el tratamiento de matices más complejos [2- 4].

La investigación de Mishra et al. (2025) revela que los enfoques tradicionales enfrentan limitaciones significativas al procesar el lenguaje periodístico, diseñado para captar la atención mediante estructuras sintácticas breves, persuasivas y emocionalmente cargadas [5]. De hecho, una de las criticas recurrentes en la literatura es la dificultad de estos modelos para manejar la ambigüedad y el encuadre sutil, especialmente en la clasificación de titulares que no son explícitamente positivos o negativos. Esta dificultad para categorizar la “neutralidad” en un dominio diseñado para no serlo, presenta una brecha de conocimiento fundamental que este trabajo busca cuantificar.

2.2.2 Sistemas existentes de análisis de medios

En el ámbito del análisis de medios de comunicación, proyectos como GDELT han impulsado la identificación de eventos y actores geopolíticos mediante el procesamiento automatizado de noticias globales [6]. No obstante, este enfoque ha estado más centrado en la detección de hechos que en la evaluación de emociones. Investigaciones recientes han comenzado a analizar millones de titulares con modelos Transformer, evidenciando tendencias longitudinales en el sentimiento expresado [6-9].

Sin embargo, todavía son escasos los desarrollos que integren arquitecturas de Big Data para el análisis de sentimiento sobre titulares de noticias a gran escala, en tiempo real, y con soporte multilingüe [5, 10]. Además, la mayoría de estos estudios aplica los paradigmas de clasificación existentes sin cuestionar su idoneidad para el dominio mediático, dejando un vacío en la investigación empírica sobre las limitaciones conceptuales de dichos enfoques.

2.3 Modelos Transformer para análisis de sentimiento multilingüe

2.3.1 Evolución de arquitecturas BERT y justificación de selección

Con la evolución del procesamiento del lenguaje natural, los modelos basados en arquitecturas Transformer, como BERT, RoBERTa, han demostrado una mayor capacidad para captar la polaridad emocional con precisión y robustez, especialmente en entornos multilingües o con estructuras lingüísticas más complejas [11-13].

La selección de modelos para este proyecto se fundamentó en criterios de eficiencia computacional, capacidad multilingüe y especialización. Considerando las limitaciones de hardware (sistema con 16GB de RAM sin GPU dedicada), se priorizaron modelos que ofrecieran un equilibrio óptimo entre precisión y recursos. Adicionalmente, la selección de un abanico de

modelos Transformer (desde el ligero DistilBERT hasta el robusto XLM-RoBERTa) se justifica como un diseño experimental para poner a prueba una hipótesis clave: la robustez inherente de estas arquitecturas frente al "ruido" textual, permitiendo investigar si las complejas fases de preprocesamiento siguen siendo tan cruciales como en épocas anteriores.

2.3.2 DistilBERT: eficiencia sin comprometer precisión

DistilBERT fue seleccionado por su excelente balance entre rendimiento y eficiencia. Según Sanh et al. (2019), mantiene el 97% de las capacidades de BERT mientras es un 40% más pequeño y un 60% más rápido [2, 5], lo que lo convierte en una opción ideal para sistemas con restricciones de recursos [4].

2.3.3 XLM-RoBERTa: capacidades multilingües avanzadas

XLM-RoBERTa fue incorporado por su superioridad en análisis multilingüe. Conneau et al. (2020) demuestran que este modelo alcanza un alto rendimiento en múltiples idiomas, superando consistentemente a modelos alternativos en tareas de clasificación de texto [14-16].

2.3.4 FinBERT: especialización en dominio financiero

FinBERT fue incluido para contrastar el rendimiento de un modelo generalista frente a uno altamente especializado. Araci (2020) demuestra que FinBERT, entrenado específicamente en corpus financieros, logra una alta precisión en su dominio [17, 18], lo que permite evaluar los efectos del "sobreajuste de dominio".

2.3.5 RoBERTuito: optimización para español

RoBERTuito fue seleccionado como modelo de referencia para el español. Pérez et al. (2022) demuestran que este modelo supera a otras alternativas en el análisis de texto de redes sociales en español, corroborando su precisión superior en detección de emociones, odio e ironía en textos en español [19, 20].

2.3.6 VADER: eficiencia léxica para procesamiento en tiempo real

VADER (Valence Aware Dictionary and sEntiment Reasoner) fue incorporado como modelo léxico de referencia por su extrema eficiencia computacional y su efectividad probada en textos breves. Hutto y Gilbert (2014) demuestran que VADER es especialmente efectivo para análisis de sentimiento en redes sociales y textos cortos como titulares [3], sirviendo como una base de comparación frente a los modelos Transformer.

2.4 Arquitecturas Big Data para análisis en tiempo real

Para manejar grandes volúmenes de noticias, utilizaremos MongoDB. Para complementarlo, la literatura apunta a soluciones como Redis para la optimización de caché, que proporciona latencias de submilisegundo [10, 21], y ElasticSearch para la indexación y consulta de millones de documentos con visibilidad inmediata [21, 22]. Este proyecto se inspira en estos principios para diseñar una arquitectura distribuida y optimizada.

2.5 Metodologías de validación y evaluación humana

La validación de sistemas de análisis de sentimiento requiere una comparación rigurosa con la evaluación humana. Estudios recientes muestran una correlación significativa entre las puntuaciones automáticas y las humana, aunque también señalan la variabilidad y subjetividad inherentes al etiquetado manual [23-26]. Dada la ambigüedad del lenguaje periodístico, especialmente en la clase neutra, se hace indispensable contar con un corpus de gran escala validado por humanos. Por ello, este proyecto adopta la metodología sHumaN (Sentimiento HUMAno incluido Neutral), con 120.000 titulares (60.000 en inglés y 60.000 en español), no solo para medir la precisión, sino para disponer de una base empírica sólida que permita investigar las discrepancias sistemáticas entre la inteligencia artificial y la humana.

Capítulo 3. OBJETIVOS

3.1 Objetivos generales

Los objetivos generales de este proyecto son:

- Desarrollar un sistema modular y escalable basado en tecnologías Big Data y Procesamiento de Lenguaje Natural (NLP) para el análisis en tiempo real del sentimiento asociado a titulares de noticias
- Encontrar y validar las fuentes de datos apropiadas para la ingesta de información que alimenten a la solución
- Estudiar e implementar las tecnologías de visualización más adecuadas para la exploración interactiva de los resultados
- Evaluar empíricamente, a través del sistema desarrollado, las limitaciones de los paradigmas actuales de análisis de sentimiento cuando se aplican al complejo dominio de los titulares de noticias

3.2 Objetivos específicos

Para alcanzar los objetivos generales, se definen los siguientes objetivos específicos, agrupados por áreas de trabajo:

Desarrollar un sistema modular y escalable basado en tecnologías Big Data

- Diseñar e implementar una arquitectura desacoplada basada en microservicios contenedorizados (Docker), que faciliten la integración, el mantenimiento y la escalabilidad del sistema.
- Construir servicios especializados para la limpieza y normalización avanzada de titulares, mejorando la calidad de los datos para el análisis posterior.

Ingesta y enriquecimiento de datos:

- Identificar y seleccionar al menos cuatro fuentes de noticias digitales con APIs públicas o gratuitas para una ingesta periódica y automatizada.
- Desarrollar un servicio de enriquecimiento geográfico automático para contextualizar especialmente los datos.

Implementación y validación de modelos de NLP

- Implementar y orquestar un conjunto híbrido de modelos de análisis de sentimiento que combinen enfoques léxicos (VADER) y modelos profundos basados en Transformers multilingües (DistilBERT, XLM-RoBERTa, FinBERT, RoBERTuito).
- Validar el rendimiento de los modelos automáticos frente a un corpus de gran escala (120.000 titulares etiquetados manualmente).

Investigación científica y análisis de resultados

- Cuantificar el impacto real del preprocesamiento de texto avanzado en el rendimiento de los modelos Transformer modernos, para validar o refutar la necesidad de pipelines de limpieza complejos.
- Analizar y medir la capacidad de los modelos de sentimiento para clasificar correctamente la categoría "neutra" en titulares de noticias, identificando patrones de error sistemáticos y sus posibles causas.

Visualización de resultados

- Construir un dashboard interactivo con Streamlit, que permita visualización dinámica y el filtrado por país, fuente, idioma y tipo de sentimiento, optimizado para consultas en tiempo real.

3.3 Beneficios del proyecto

Los beneficios del proyecto se clasifican en tres áreas principales

- **Académicos y científicos**:
  - Proporciona una solución innovadora para análisis emocional de titulares, adaptable a múltiples idiomas y dominios.
  - Aporta evidencia empírica sobre la "Paradoja del Preprocesamiento" y el "Problema de la Neutralidad", contribuyendo al debate sobre la aplicabilidad de las técnicas estándar de NLP en dominios complejos.
  - Facilita la validación empírica a gran escala mediante un corpus de 120.000 noticias, contribuyendo a la literatura sobre análisis de sentimiento en medios digitales.
- **Tecnológicos**:
  - Ofrece una arquitectura modular y escalable que permite una integración sencilla y un despliegue flexible en entornos locales o en la nube.
  - Demuestra un uso eficiente de recursos mediante combinación de modelos ligeros y profundos, optimizado para hardware estándar.
- **Sociales y prácticos**:
  - Apoya la comprensión del impacto emocional de los medios en la formación de opinión pública.
  - Constituye una herramienta útil para investigadores, analistas de medios, instituciones y comunicadores.
  - Promueve el uso responsable y transparente de la IA en el análisis mediático.

Figura 3.1: Beneficios del proyecto

Capítulo 4. DESARROLLO DEL PROYECTO

4.1 Planificación del proyecto

El desarrollo del proyecto se ha estructurado a lo largo de seis meses, desde febrero hasta julio de 2025, organizando el trabajo en fases consecutivas: investigación, diseño, implementación, validación, documentación y presentación. A continuación, se describen las actividades principales realizadas o previstas en cada etapa:

- **Febrero 2025 – Exploración y planteamiento inicial:**
  Se definió el alcance del proyecto, se identificó la problemática a abordar y se delimitaron los objetivos generales y específicos. Asimismo, se recopilaron fuentes bibliográficas y se realizó un primer análisis del estado del arte en técnicas de análisis de sentimiento, consumo digital de noticias y tecnologías Big Data. También se eligió el enfoque general de la arquitectura técnica.
- **Marzo 2025 – Diseño arquitectónico y selección de herramientas:**
  Se diseñó la arquitectura del sistema siguiendo una aproximación modular basada en contenedores Docker. Se identificaron las APIs públicas de noticias a utilizar (NewsAPI, GNews, NewsData, The Guardian), y se eligieron modelos de NLP adecuados (VADER, DistilBERT, etc). Se definieron los microservicios y flujos de trabajo para el procesamiento, enriquecimiento y visualización de datos.
- **Abril 2025 – Desarrollo técnico y pruebas iniciales:**
  Se implementaron los microservicios de ingesta periódica, limpieza avanzada de títulos, análisis de sentimiento automático y geolocalización. Se configuró una base de datos MongoDB como almacenamiento central, y se construyó el dashboard interactivo con Streamlit. Se realizaron pruebas unitarias y de integración en entorno local utilizando Docker Compose.
- **Mayo 2025 – Validación funcional y evaluación comparativa:**
  Se ejecutaron pruebas de rendimiento, robustez y calidad de los resultados. Se incorporó una muestra de titulares evaluados manualmente por humanos para comparar el desempeño de los modelos automáticos (VADER, DistilBERT, etc). Se ajustaron reglas de limpieza y se documentaron los resultados preliminares. Se definieron métricas de evaluación como exactitud, polaridad y coherencia del análisis.
- **Junio 2025 – Redacción y consolidación de la memoria del TFM:**
  Durante este mes se consolidaron todos los capítulos de la memoria, incorporando el análisis de resultados, capturas del sistema, referencias bibliográficas y diagramas técnicos. Se preparan los anexos y se integran las evidencias necesarias para respaldar el trabajo realizado.
- **Julio 2025 – Preparación final y defensa del proyecto:**
  Se elabora la presentación oficial del TFM, con visualizaciones relevantes, explicación detallada de la solución propuesta y resumen de los resultados obtenidos. Se ensayó la defensa, se revisó la documentación completa y se entregó el documento final en los plazos establecidos por la universidad.

Este plan ha seguido un enfoque iterativo y flexible, permitiendo ajustes progresivos según el avance del desarrollo técnico, garantizando al mismo tiempo una entrega ordenada y completa.

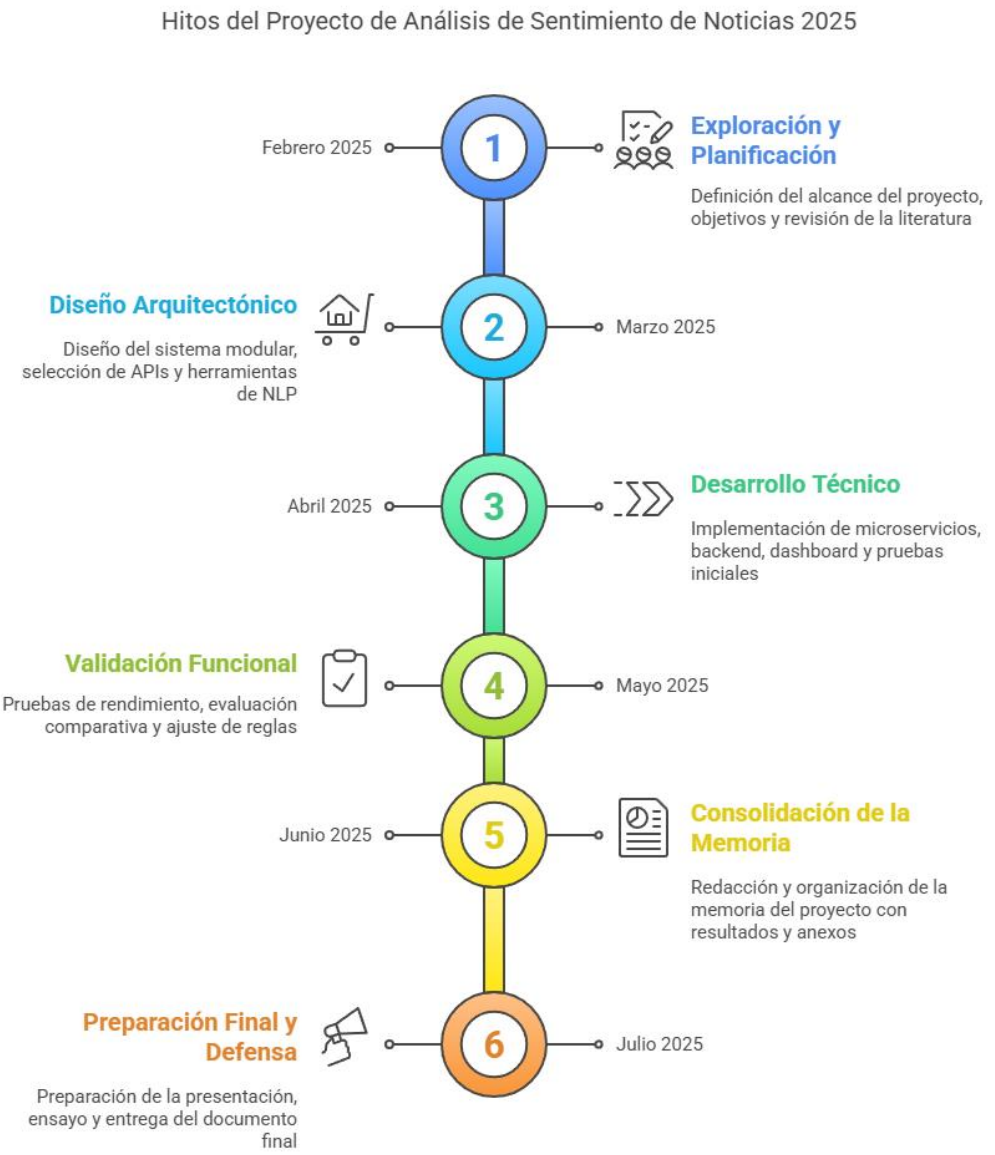


Figura 4.1: Diagrama de Gantt del proyecto

Cronograma de desarrollo del sistema de análisis de sentimiento sobre titulares de noticias. Se representan las seis fases consecutivas realizadas entre febrero y julio de 2025: exploración, diseño, desarrollo técnico, validación, redacción y defensa final.

4.2 Descripción de la solución, metodologías y herramientas empleadas

4.2.1 Metodologías del proyecto

El enfoque metodológico adoptado combina una perspectiva tecnológica aplicada, integrando software libre, contenedores Docker y modelos NLP con aplicaciones prácticas, con un análisis cuantitativo basado en datos estructurados y métricas evaluables, además de una validación empírica sustentada en clasificación manual de muestras reales. Este enfoque hibrido garantiza tanto la funcionalidad del prototipo como la evaluación objetiva de su rendimiento.

4.2.2 Solución técnica desarrollada

La arquitectura modular se ha implementado mediante Docker Compose, asegurando portabilidad, escalabilidad y facilidad de despliegue. Los microservicios se organizan en bloques funcionales específicos para la ingesta de datos por idioma, limpieza y normalización de titulares, enriquecimiento geográfico, análisis de sentimiento y visualización. Este diseño permite una gestión eficiente y mantenimiento independiente de cada componente.

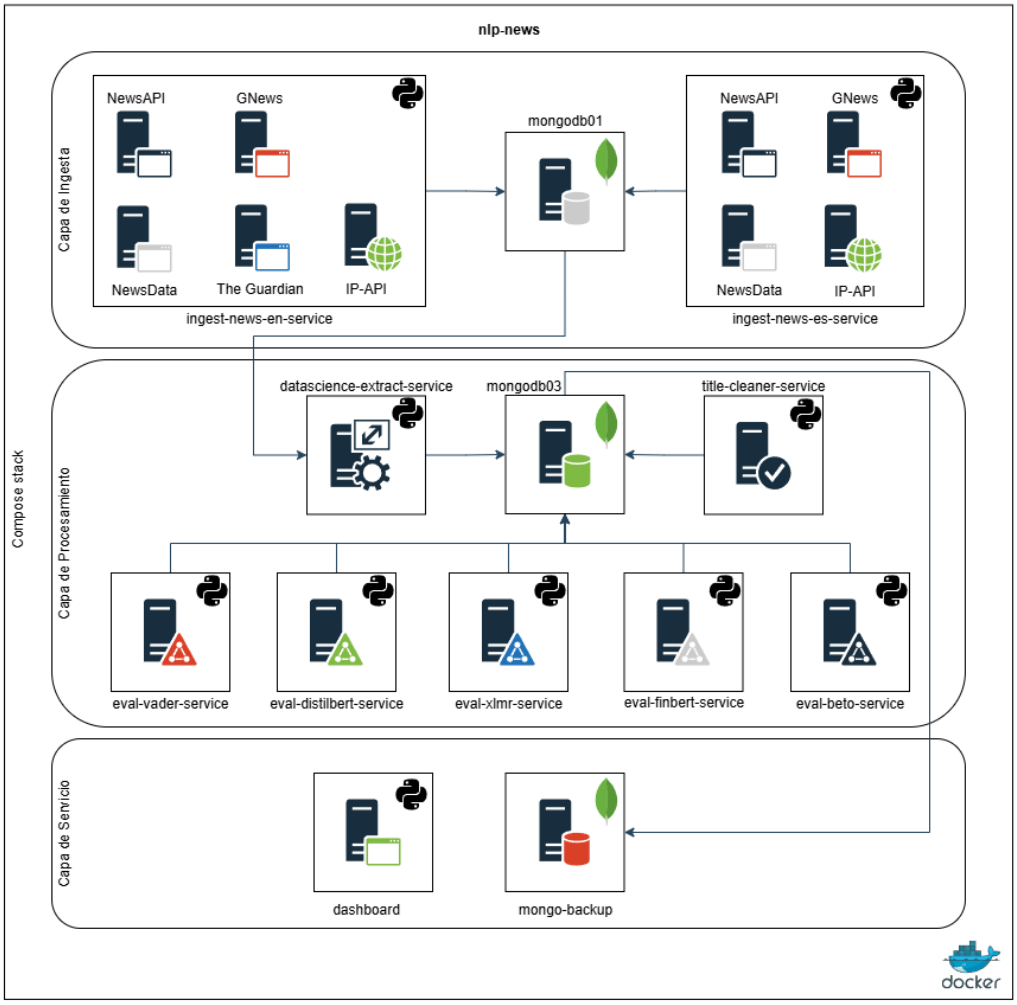


Figura 4.2: Arquitectura desarrollada

Los principales bloques funcionales son:

Ingesta de datos por idioma:

Dos servicios principales se encargan de la ingesta de titulares: **ingest-news-en-service** para inglés y **ingest-news-es-service** para español. Cada uno integra múltiples fuentes (NewsAPI, GNews, NewsData.io, The Guardian) y ejecuta recolecciones periódicas. Los datos se almacenan en una base de datos NoSQL, de tipo MongoDB (ver Figura 4.2 recuadro mongodb01).

Especificaciones técnicas de las fuentes de datos

El sistema integra cuatro APIs principales para la recolección de noticias. A continuación, se detallan las especificaciones técnicas de cada fuente:

Campo	Valor
Nombre de la fuente	NewsAPI
URL del endpoint	https://newsapi.org/v2/top-headlines
Formato de datos	JSON
Idiomas soportados	en, es
Método HTTP	GET
Parámetros de consulta (inglés)	apiKey: NEWSAPI_KEY language: en pageSize: 10 page: 1
Parámetros de consulta (español)	apiKey: NEWSAPI_ES_KEY language: es q: a pageSize: 100 page: 1
Campos almacenados	source author title description

	url urlToImage publishedAt content datetime (metadato propio) apiSource (metadato propio)
Campos/datos descartados	Ninguno
Colección MongoDB (inglés)	news
Colección MongoDB (español)	noticias
Límites de la API	Plan gratuito: 100 solicitudes/día
Frecuencia	cada 15 minutos
Endpoint español	https://newsapi.org/v2/everything
Control de duplicados	Por campo url

Tabla 4.1: Especificación de fuente de datos NewsAPI

Campo	Valor
Nombre de la fuente	GNews
URL del endpoint	https://gnews.io/api/v4/top-headlines
Formato de datos	JSON
Idiomas soportados	en, es
Método HTTP	GET
Parámetros de consulta (inglés)	token: GNEWS_API_KEY lang: en max: 10 page: 1
Parámetros de consulta (español)	lang: es

	max: 10 token: GNEWS_ES_KEY
Campos almacenados	title description content url image publishedAt source datetime apiSource
Campos/datos descartados	Artículos sin URL
Colección MongoDB (inglés)	news
Colección MongoDB (español)	noticias
Límites de la API	Plan gratuito: 100 solicitudes/día
Frecuencia	cada 15 minutos
Límite por request	Inglés: 10 artículos Español: 10 artículos
Control de duplicados	Por campo url

Tabla 4.2: Especificación de fuente de datos GNews

Campo	Valor
Nombre de la fuente	NewsData
URL del endpoint	https://newsdata.io/api/1/news
Formato de datos	JSON
Idiomas soportados	en, es
Método HTTP	GET

Parámetros de consulta (inglés)	apikey: NEWSDATA_API_KEY language: en
Parámetros de consulta (español)	apikey: NEWSDATA_ES_KEY language: es
Campos almacenados	article_id title link keywords description pubDate pubDateTZ image_url source_id source_name source_priority source_url source_icon language country category datetime apiSource
Campos/datos descartados	Articulos sin link video_url (sólo en español)
Colección MongoDB (inglés)	news
Colección MongoDB (español)	noticias
Límites de la API	Plan gratuito: 200 solicitudes/día
Frecuencia	cada 8 minutos

Control de duplicados	Por campo link

Tabla 4.3: Especificación de fuente de datos NewsData

Campo	Valor
Nombre de la fuente	The Guardian
URL del endpoint	https://content.guardianapis.com/search
Formato de datos	JSON
Idiomas soportados	en
Método HTTP	GET
Parámetros de consulta	api-key: THEGUARDIAN_API_KEY show-fields: tailText, body page-size: 200 page: variable (1-5)
Campos almacenados	id type sectionId sectionName webPublicationDate webTitle webUrl apiUrl fields isHosted pillarId pillarName datetime apiSource
Campos/datos descartados	Valores None y objetos dict vacíos

Colección MongoDB	news
Límites de la API	5.000 solicitudes/día
Frecuencia	cada 5 minutos
Paginación	Hasta 5 páginas por consulta (200 artículos/página)
Campos especiales	fields.trailText, fields.body
Control de duplicados	Por campo webUrl

Tabla 4.4: Especificación de fuente de datos The Guardian

Limpieza y normalización:

El proceso de limpieza y normalización constituye una etapa crítica que transforma los títulos originales (title) en versiones optimizadas para análisis de sentimiento (cleanTitle).

El servicio **title-cleaner-service** implementa un pipeline de nueve operaciones secuenciales (Ver Figura 4.4), ejecutadas en el servidor de procesamiento de datos (Ver Figura 4.2 recuadro mongodb03), procesando lotes de 1.000 documentos cada 10 minutos.

Este diseño exhaustivo no solo busca mejorar la calidad de los datos, sino que fue concebido de manera deliberada como parte del diseño experimental para poner a prueba uno de los objetivos científicos del proyecto: cuantificar el impacto real de un preprocesamiento intensivo en el rendimiento de los modelos Transformer modernos, sentando las bases para analizar la "Paradoja del Preprocesamiento".

Campo	Valor
Nombre del servicio	Title Cleaner Service
Servidor de ejecución	mongodb03 (servidor de ciencia de datos)
Frecuencia de procesamiento	Cada 10 minutos
Tamaño de lote	1000 documentos por lote
Fuente de datos	Vista especializada (EVAL_VIEW_NAME)
Campo de entrada	title (título original)
Campo de salida principal	cleanTitle (título procesado)

Campo de control de calidad	invalidLang (detección de idiomas no soportados)
Tecnología de normalización	Unicode NFKC, expresiones regulares, demoji
Control de errores	Por documento individual y global

Tabla 4.5: Detalles técnicos del servicio de limpieza

Operaciones de limpieza secuenciales

El sistema aplica las siguientes transformaciones en orden específico, garantizando la preservación del contenido semántico mientras elimina elementos que podrían interferir con el análisis de sentimiento:

#	Operación	Descripción Técnica	Ejemplo de Transformación
1	Normalización Unicode NFKC	unicodedata.normalice("NFKC", title)	Unifica caracteres con representaciones múltiples
2	Eliminación de caracteres invisibles	Remover espacios de ancho cero, marcadores bidireccionales	"Noticiaimportante" → "Noticia importante"
3	Eliminación de emojis	demoji.replace(title, "")	"Gran victoria 🎉🏆 del equipo nacional ⚽" → "Gran victoria del equipo nacional"
4	Normalización de caracteres especiales	Conversión de caracteres de reemplazo y viñetas	"Crisis – económica" → "Crisis – económica"
5	Estandarización de separadores	Unificar " – ", " — " → " – "	"Análisis – perspectivas — futuras" → "Análisis - perspectivas – futuras"
6	Compactación de separadores	Reducir separadores repetidos a uno solo	"Noticia - - - importante" → "Noticia – importante"
7	Normalización de espacios	re.sub(r'\s+', ' ', title).strip()	" Título con espacios " → "Título con espacios"

8	Eliminación de sufijos de fuente	Comparación con base de datos de medios conocidos	"Mercados suben tras anuncio – Reuters" → "Mercados suben tras anuncio"
9	Detección de idiomas inválidos	Identificación de caracteres árabes/chinos	"Breaking news الأخبار العاجلة" → Marca invalidLang=true

Tabla 4.6: Operaciones del servicio de limpieza

Algoritmo inteligente de eliminación de sufijos

Una de las innovaciones más significativas del sistema es la eliminación automática de nombres de medios que aparecen como sufijos redundantes en los títulos. El algoritmo funciona de la siguiente manera:

Proceso de detección:

1. División por separador estándar: el titulo se segmenta usando " - " como delimitador
2. Validación de longitud: el sufijo debe contener entre 1-5 palabras para ser considerado válido
3. Normalización alfanumérica: se eliminan caracteres no alfanuméricos del sufijo para comparación.
4. Consulta en base de datos: se compara contra la base precargada de medios conocidos
5. Eliminación selectiva: si coincide, se remueve solo la última parte, preservando el contenido principal

Base de datos de referencia precargada:

- Fuentes conocidas: nombres oficiales extraídos de processed_media
- Dominios registrados: URLs de fuentes de noticias
- Aliases: nombres alternativos y variaciones

title	cleanTitle
Inflación alcanza nuevo máximo histórico – El País	Inflación alcanza nuevo máximo histórico
Tesla registra pérdidas trimestrales – Reuters	Tesla registra pérdidas trimestrales
Nuevo récord de temperatura global – BBC News	Nuevo récord de temperatura global

Tabla 4.7: Ejemplos de transformaciones reales

Sistema de control de calidad y detección de idiomas

El servicio incorpora un sistema automático de control de calidad que identifica longitudes que exceden la capacidad y títulos en idiomas no soportados por los modelos de análisis de sentimiento (Ver Tabla 4.8 de ejemplos de casos detectados).

Criterios de invalidación:

- Detección de árabe: rango Unicode [\u0600-\u06FF]
- Detección de chino: rango Unicode [\u4E00-\u9FFF]
- Detección de longitud: mayor a 512 caracteres
- Marcado automático: campo invalidLang=true para exclusión de análisis

title	invalidLang
Important news about الأخبار العاجلة today	true
Financial report 中国新闻 análisis	true

Tabla 4.8: Ejemplos de casos detectados

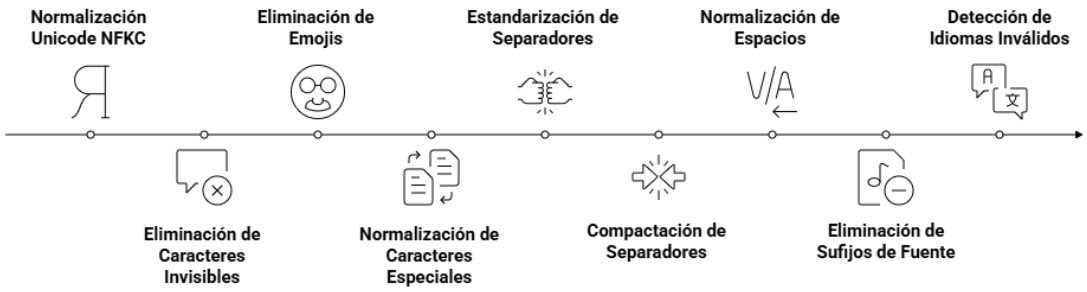


Figura 4.3: Pipeline title-cleaner

Enriquecimiento geográfico:

El servicio **ingest-geolocation-service** asigna datos geográficos a cada noticia a partir del dominio o fuente, utilizando la base de datos media y la API pública de geolocalización IP-API [28].

Campo	Valor
Nombre del sistema	Media Registry & Geolocation Service
APIs utilizadas	IP-API (http://ip-api.com)
Método de extracción	tldextract para dominios registrados
Formato de entrada	URLs de noticias de todas las fuentes
Formato de salida	Documentos MongoDB con geolocalización
Campos de geolocalización	country (código ISO)

	city (nombre ciudad) coordinates [longitud, latitud]
Colecciones MongoDB	media (inglés) medios (español)
Estructura del documento	_id: ID único slugificado name: Nombre del medio domain: Dominio registrado aliases: Array de nombres alternativos wLocations: Array de ubicaciones web
Límites de API	40 solicitudes/minuto a IP-API
Frecuencia	cada 5 minutos
Control de duplicados	Por dominio y coordenadas geográficas
Manejo de errores	Logs específicos, fallback para dominios sin geolocalización
Campo de control	evaluatedMedia en documentos de noticias

Tabla 4.9: Especificación técnica de Sistema de Geolocalización

Análisis de sentimiento:

Se aplican varios modelos complementarios para cubrir diferentes idiomas y contextos.

Esta estrategia multimodelo es fundamental para el diseño de la investigación, ya que permite una evaluación comparativa rigurosa. Al enfrentar diferentes arquitecturas (léxicas, Transformer generalistas, especializadas en idioma y en dominio) contra el mismo dataset, se crea el entorno controlado necesario para investigar el "Problema de la Neutralidad" y determinar si es un fallo de un modelo específico o una limitación sistemática del paradigma actual.

- eval-vader-service: modelo léxico basado en reglas para textos breves [3].
- eval-finbert-service: modelo especializado en textos financieros [17].
- eval-beto-service: (RoBERTuito) es un pequeño RoBERTa entrenado en español [19].
- eval-distilbert-service: versión reducida de BERT multilingüe [2].
- eval-xlmr-service: modelo multilingüe robusto basado en RoBERTa [7, 14].

Almacenamiento distribuido:

Los datos se guardan en MongoDB 5.0 distribuidos en dos servidores: mongodb01 para datos crudos y mongodb03 para datos procesados y estructurados para análisis y visualización.

Esta separación, además de optimizar el rendimiento al aislar las cargas de lectura y escritura, cumple una función metodológica: segrega el repositorio de datos brutos del entorno de análisis, garantizando que los experimentos se realicen sobre un conjunto de datos procesados y consistente.

Nombre	Tipo	Descripción
ingest-news	Base de datos	Contiene información obtenida a través de las APIs
news	Colección	Contiene noticias de la ingesta en inglés
noticias	Colección	Contiene noticias de la ingesta en español
media	Colección	Contiene medios de la ingesta en inglés
medios	Colección	Contiene medios de la ingesta en español
NewsWithoutMedia	Vista	Contiene las noticias que aún no están asociadas a un medio

Tabla 4.10: Contenido mongodb01

Nombre	Tipo	Descripción
extract-news	Base de datos	Contiene información procesada para el análisis de datos
processed_news	Colección	Contiene las noticias procesadas
processed_media	Colección	Contiene los medios procesados
pending_beto	Vista	Documentos pendientes de evaluar por RoBERTuito
pending_distilbert	Vista	Documentos pendientes de evaluar por DistilBERT
pending_vader	Vista	Documentos pendientes de evaluar por VADER
pending_xlmroberta	Vista	Documentos pendientes de evaluar por XLMR
pending_title_clean	Vista	Documentos pendientes por limpieza de títulos
news_with_location	Vista	Noticias geolocalizadas

Tabla 4.11: Contenido mongodb03

Optimización de acceso a datos:

Se crearon vistas (ver Tablas 4.10 y 4.11) e índices sobre campos clave para optimizar el rendimiento (ver Anexos 10 y 11), durante la ingesta, procesamiento, evaluación y visualización; sin acceder directamente a las colecciones.

Nombre	Colección	Uso	Propiedades
_id_	media	>50	Unique
domain_1	media	>4000	
aliases_1	media	>500	
_id_	medios	>20	Unique
aliases_1	medios	>100	
domain_1	medios	>3000	
_id_	news	>3000	Unique
evaluatedMedia.checkedAt_1	news	>100	
_id_	noticias	>3000	Unique
evaluatedMedia.checkedAt_1	noticias	>100	

Tabla 4.12: Índices principales mongodb01

Nombre	Colección	Uso	Propiedades
_id_	processed_media	>500K	Unique
domain_1	processed_media	>500K	
_id_	processed_news	>3M	Unique
domain_1	processed_news	>15K	

Tabla 4.13: Índices principales mongodb03

Colección	Índice	COLSCAN (ms)	IXSCAN (ms)
ingest-news.media	domain_1	2	0

ingest-news.news	evaluatedMedia.checkedAt_1	243	6
extract-news.processed_media	domain_1	5	2
extract-news.processed_news	extractedAt_-1	268	217

Tabla 4.14: Comparación planes de ejecución

Visualización interactiva:

Para la exploración y el análisis de datos procesados, se ha desarrollado un módulo de visualización interactiva en forma de aplicación web. Esta herramienta fue construida utilizando Streamlit, un framework de Python de código abierto diseñado para crear y compartir aplicaciones de ciencia de datos de manera más rápida y sencilla.

El resultado es un Dashboard analítico multipágina que permite a los usuarios interactuar con los datos en tiempo real, aplicar filtros dinámicos y obtener insights visuales a través de diversas representaciones gráficas.

La aplicación se estructura en varias secciones, accesibles a través de un panel de navegación lateral. Este panel también contiene un conjunto de filtros globales que se aplican de manera persistente en todas las páginas de la aplicación, garantizando una experiencia de usuario cohesiva y un análisis consistente (Ver Anexo 1). Los filtros disponibles son:

- Idioma: permite seleccionar el idioma de las noticias a analizar
- Modelo (sentimiento): ofrece la posibilidad de elegir el modelo de análisis de sentimiento aplicado (ej. XLM-RoBERTa)
- Fuente / Dominio: permite acotar el análisis a un medio de comunicación específico (ej. theguardian.com)
- Rango de fechas: selecciona un intervalo de tiempo específico para el análisis
- País: filtra las noticias según su país de origen

A continuación, se describen las diferentes páginas del Dashboard:

Portada: Últimas Noticias

Esta sección funciona como la página de inicio y ofrece una vista rápida de las noticias más recientes que han sido procesadas por el sistema. Se presenta en forma de listado, mostrando para cada noticia:

- El titular (con un enlace al artículo original, que permite hacer clic para abrirse en una pestaña/ventana nueva)
- La fuente y el dominio del medio
- La fecha y hora de publicación

Esta vista es fundamental para verificar la ingesta de datos en tiempo real y tener un primer contacto con los temas de actualidad (Ver Anexo 2).

Inicio: KPIs y Resumen General

La página de "Inicio" constituye el cuadro de mando principal, proporcionando una visión general y agregada de los datos a través de varios componentes visuales:

- Mapa de Noticias en Tiempo Real: un mapa mundial interactivo que geolocaliza las noticias, permitiendo identificar focos de actividad informativa de un solo vistazo.
- Indicadores Clave (KPIs): tarjetas que resumen las métricas más importantes del conjunto de datos filtrado:
    - Noticias Totales: el número total de artículos que cumplen los criterios de los filtros.
    - Países Detectados: el recuento de países únicos de los que provienen las noticias.
    - Días con Actividad: el número de días únicos con registros de noticias.
- Distribución de Sentimiento: un gráfico de barras que muestra el volumen de noticias clasificadas como positivas, neutras y negativas.
- Top 10 Países: un gráfico de barras que clasifica los 10 países con mayor volumen de noticias, facilitando la comparación de la cobertura mediática por región.
- Volumen de Noticias a lo Largo del Tiempo: un gráfico de líneas que representa la evolución del número de noticias publicadas día a día, permitiendo identificar tendencias, picos de actividad y patrones temporales (Ver Anexo 3-5).

Palabras y Clústeres

Esta sección se adentra en el análisis textual de los titulares de las noticias para identificar los temas y conceptos más relevantes.

- Análisis de Palabras Clave:
    - Top palabras (global): un gráfico de barras muestra las palabras más frecuentes en todo el conjunto de datos filtrado.
    - Palabras por Sentimiento: tres tablas detallan las palabras más frecuentes en todo el conjunto de datos filtrado.
- Clúster de palabras (títulos): una visualización avanzada que representa las palabras como puntos en un espacio de dos dimensiones. Las palabras semánticamente similares aparecen agrupadas, y el color de los puntos indica el clúster al que pertenecen. Un deslizador permite ajustar el número de clústeres a generar, facilitando la exploración de agrupaciones temáticas emergentes (Ver Anexo 6, 7).

Explorador de Noticias

Esta es una potente herramienta de búsqueda que permite al usuario realizar consultas directas sobre la base de datos de noticias. Introduciendo una palabra o frase, el sistema devuelve un listado paginado con todos los artículos que coinciden con la búsqueda. La tabla de resultados incluye el título, la URL del artículo original, el dominio y la fecha de publicación, permitiendo ordenarlo por cualquiera de las columnas, descargar los resultados, realizar un análisis detallado y recuperación de información específica (Ver Anexo 8).

En conjunto, este Dashboard interactivo no solo presenta los datos de una manera clara y accesible, sino que también empodera al usuario para que explore, filtre y descubra insights por sí mismo, pasando de una visión macro (mapa y KPIs) a una visión micro (explorador de noticias) de forma fluida e intuitiva.

Módulo de visualización	Propósito y funcionalidad
Panel de Filtros Globales	Permite al usuario segmentar y explorar el conjunto de datos de noticias de manera global y persistente. Ofrece filtros por idioma, modelo de sentimiento, polaridad (positivo, neutro, negativo), fuente/dominio, rango de fechas y país, que se aplican a todas las vistas de la aplicación. (Anexo 1)
Portada: Últimas Noticias	Actúa como pantalla de bienvenida, ofreciendo una vista dinámica de los artículos más recientes procesados por el sistema. Muestra el titular, la fuente y la fecha de publicación, permitiendo al usuario verificar la ingesta de datos en tiempo real. (Anexo 2)
Inicio: KPIs y Resumen General	Proporciona una visión agregada y de alto nivel del panorama informativo. Integra múltiples componentes: un mapa mundial para geolocalizar noticias, **Indicadores Clave (KPIs)**, gráficos de **distribución de sentimiento** y **Top 10 países**, y una gráfica de **volumen de noticias a lo largo del tiempo** para identificar tendencias. (Anexo 3-5)
Palabras y Clústeres	Se enfoca en el análisis textual de los titulares para identificar temas y conceptos clave. Presenta las **palabras más frecuentes** a nivel global y desglosadas por sentimiento. Incluye una visualización de **clústeres de palabras** que agrupa términos semánticamente similares para descubrir relaciones temáticas. (Anexo 6, 7)
Explorador de Noticias	Ofrece una funcionalidad de búsqueda detallada para la recuperación de información específica. Permite al usuario buscar noticias por palabra clave o frase y devuelve una **tabla paginada** con los resultados, incluyendo el título, el enlace al artículo original, el dominio y la fecha. (Anexo 8)

Tabla 4.15: Módulos de visualización

Control de usuarios MongoDB:

Se implementaron perfiles con distintos permisos para controlar accesos y mantener la seguridad.

Usuario	Roles	Función
admin_backup	restore backup readWriteAnyDatabase	gestión de respaldos de las bases de datos
sa	root	administración de las bases de datos
user_editor	readWrite	lectura y escritura sobre ingest-news
user_reader	read	lectura sobre ingest-news

Tabla 4.16: Usuarios en mongodb01

Usuario	Roles	Función
admin_backup	restore backup readWriteAnyDatabase	gestión de respaldos de las bases de datos
sa	root	administración de las bases de datos
user_editor	readWrite	lectura y escritura sobre extract-news
user_reader	read	lectura sobre extract-news

Tabla 4.17: Usuarios en mongodb03

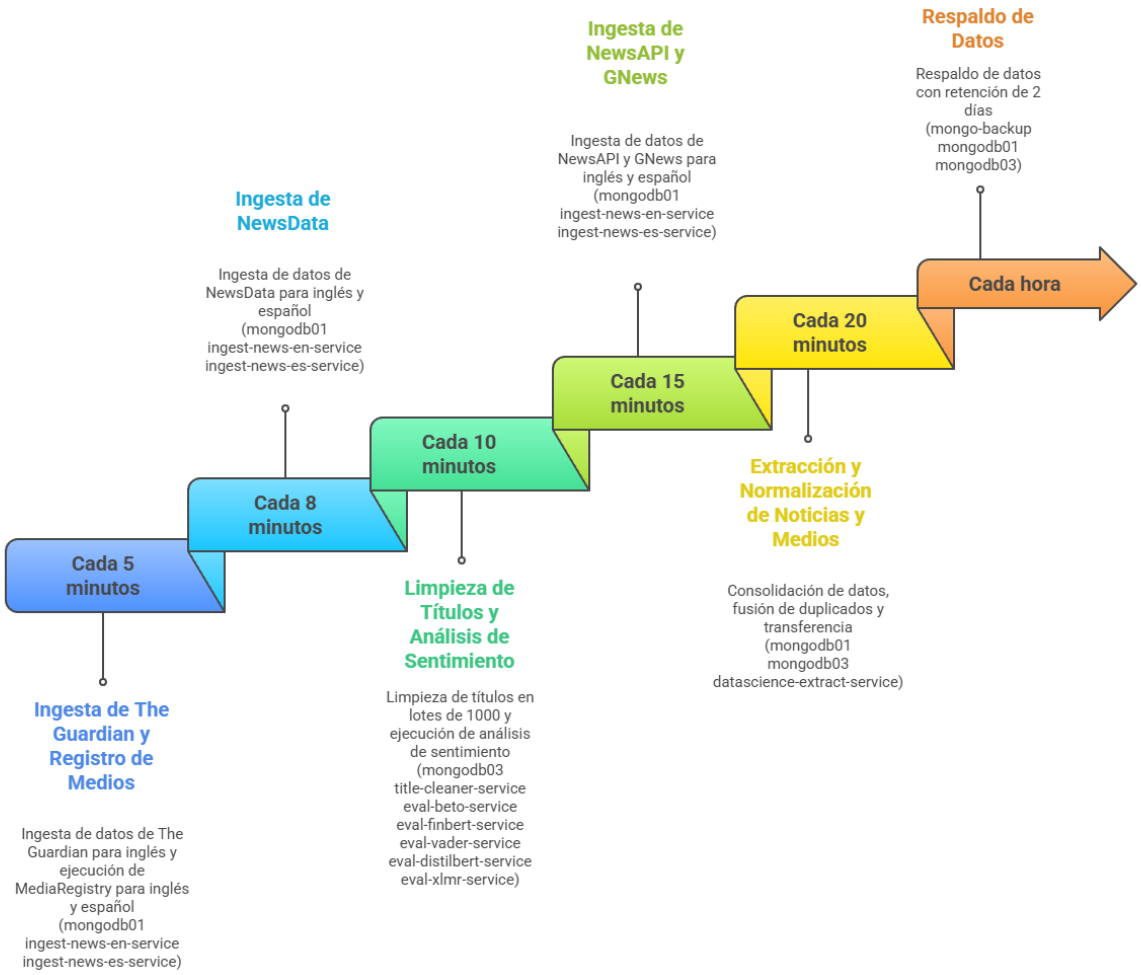


Figura 4.4: Flujo de orquestación y tolerancia a fallos

4.3 Recursos requeridos

A continuación, se enumeran los recursos necesarios para la ejecución y desarrollo del proyecto:

Hardware

Nombre	Descripción
Equipo de desarrollo (ordenador personal o portátil)	Procesador: Intel Core i7 / AMD Ryzen 7 (generación 2022 o posterior) Memoria RAM: 32 GB (mínimo recomendable: 16 GB) Almacenamiento: SSD de 512 GB (óptimo: 1 TB) Pantalla: Monitor de 24” con resolución Full HD o superior Conectividad: Acceso a internet de fibra óptica (>300 Mbps)

Tabla 4.18: Requisitos de hardware

Software

Nombre	Descripción
Sistema operativo	Linux (Ubuntu 22.04) o Windows 11
Contenedores y orquestación	Docker + Docker Compose
Lenguaje de programación	Python 3.10
Librerías y frameworks	FastAPI, Uvicorn, Pydantic Transformers, Torch, NLTK, Regex, HTTPX, Motor (cliente MongoDB para Python) Streamlit (visualización interactiva)
Entorno de bases de datos	MongoDB 5.0
Servicios y herramientas de terceros	APIs públicas de noticias: NewsAPI, GNews, NewsData.io, The Guardian API de geolocalización: IP-API
Plataforma de desarrollo colaborativo y control de versiones	Git y GitHub
Ofimática	Microsoft 365

Tabla 4.19: Requisitos de software

Recursos digitales y de infraestructura

- Cuenta de Microsoft 365 (para la documentación)
- Conexión a internet con disponibilidad 24/7 durante la fase de pruebas y validación
- Acceso a plataformas de documentación técnica y repositorios de modelos pre-entrenados (Hugging Face)
- Espacio en disco suficiente para el almacenamiento temporal y persistente de noticias (mínimo 50 GB durante pruebas intensivas)
- Backup automático de la base de datos (opcional en fase de producción)

4.4 Presupuesto

El presupuesto total del proyecto se ha estimado considerando tanto el coste del tiempo dedicado como los recursos técnicos utilizados. A continuación, se presenta un desglose detallado:

Concepto	Valor	Descripción
Horas de trabajo en el proyecto	30.000€ (300h * 100€/h)	Estimación mínima de dedicación para un trabajo de fin de máster.
Equipo técnico utilizado	1.500€	Portátil o sobremesa con procesador i7/Ryzen 7, 32 GB RAM, SSD 1 TB, monitor 24”.
Software utilizado	99€	Salvo Microsoft 365, todo el software utilizado es de código abierto y gratuito (Python, Docker, etc.)
Servicios (internet y electricidad)	480€ (80€/mes * 6)	Coste mensual de servicio de internet y electricidad durante los meses del desarrollo.
Infraestructura en la nube	0€	Se utilizaron únicamente servicios locales y gratuitos (free tier).
Estudios e informes	0€	La bibliografía ha sido de acceso público o facilitada por la universidad.
Materiales empleados	0€	No se ha requerido el uso de materiales físicos o sensores.
Costes directos	32.079€	
Costes indirectos	480€	
Gastos generales (GG)	4.170,27€	13% de los costes directos
Subtotal con GG	36.729,27€	
Beneficio Industrial	3.672,93€	10%
Subtotal (sin IVA)	40.402,2€	Sin impuesto sobre el valor añadido
IVA	8.484,46€	Impuesto sobre el valor añadido
Total	**48.886,66€**	

Tabla 4.20: Presupuesto

4.5 Viabilidad

Viabilidad técnica

El proyecto presenta una viabilidad técnica sólida, fundamentada en el uso de tecnologías probadas y escalables. La arquitectura modular basada en microservicios desplegados en contenedores Docker permite la portabilidad y fácil mantenimiento del sistema, facilitando su despliegue en distintos entornos locales o remotos sin necesidad de grandes modificaciones.

Viabilidad económica

Desde el punto de vista económico, el coste total estimado del proyecto es de aproximadamente 48.887€, que incluye la valoración del tiempo de trabajo profesional, equipo técnico, servicios de internet y electricidad. Este coste es razonable y competitivo, considerando la especialización técnica del proyecto y su capacidad para reutilizar componentes en futuros desarrollos o aplicaciones.

El uso exclusivo de software gratuito (Python, Docker, MongoDB, etc.) elimina gastos asociados a licencias, reduciendo significativamente la inversión inicial y de mantenimiento. La infraestructura empleada es local o basada en servicios gratuitos, evitando costes en la nube que puedan encarecer la solución.

Esta relación coste-beneficio facilita la replicabilidad del proyecto en contextos académicos, de investigación y empresariales con presupuestos limitados, posicionando la solución como una alternativa accesible para análisis mediático avanzado.

Viabilidad legal y ética

El proyecto cumple con los marcos legales y normativos aplicables, incluyendo la protección de datos personales y derechos de propiedad intelectual. La ingesta de datos se realiza exclusivamente desde fuentes públicas y APIs oficiales que permiten el acceso gratuito o bajo licencia, respetando términos y condiciones de uso.

El tratamiento de datos se limita a titulares de noticias, sin recopilación de información personal sensible ni datos de usuarios finales, minimizando riesgos legales relacionados con privacidad. Además, se han implementado buenas prácticas de seguridad, como gestión segura de credenciales y roles en bases de datos, para proteger la integridad y confidencialidad del sistema.

En términos éticos, el proyecto promueve la transparencia y el uso responsable de inteligencia artificial en medios digitales, evitando sesgos y fomentando la validación humana de los resultados. La solución busca contribuir positivamente a la comprensión social de la opinión pública, sin manipulación ni uso indebido de la información.

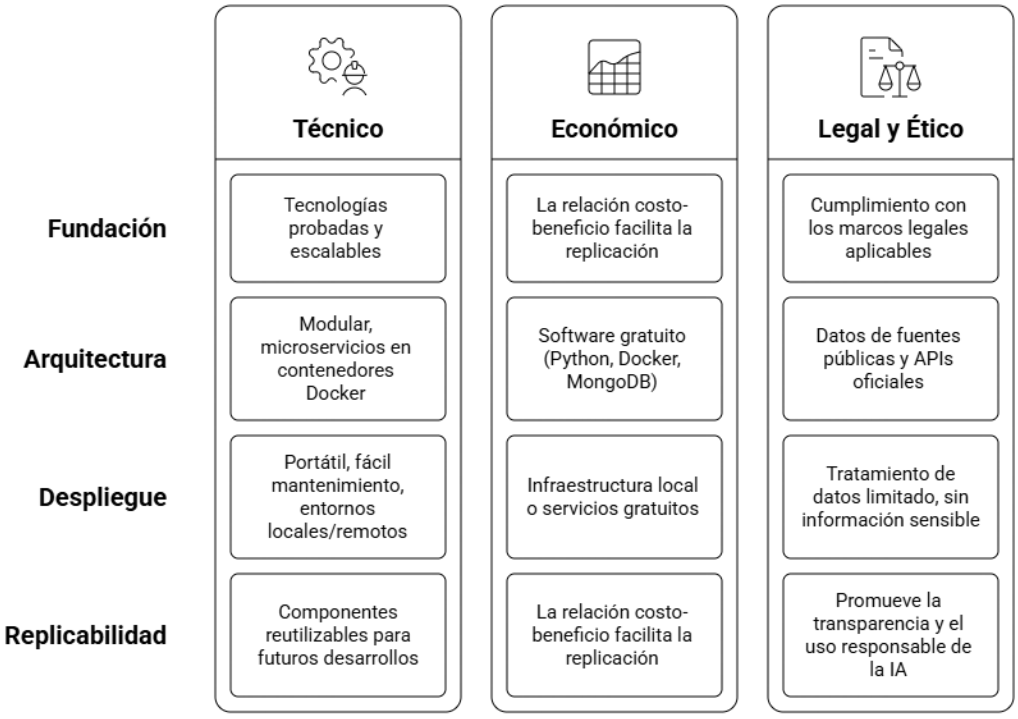


Figura 4.5: Viabilidad del proyecto

4.6 Resultados del proyecto

Los resultados del proyecto concuerdan con los objetivos planteados al principio de este trabajo al desarrollarse un sistema funcional para análisis de sentimiento en titulares en tiempo real, aplicando Big Data y NLP. La solución cumple los objetivos planteados y fue validada empíricamente.

Entre los logros técnicos destacan:

- Arquitectura completa en Docker con servicios para ingesta, limpieza, análisis, almacenamiento, visualización y geolocalización.
- Integración de múltiples fuentes de noticias con ingesta automatizada que consume más de 3.000 noticias al día.
- Servicio de limpieza avanzada para generación de cleanTitle.
- Análisis de sentimiento con modelos ligeros (VADER) y profundos (DistilBERT, XLM-RoBERTa).
- Geolocalización automática mediante dominio/fuente y API IP-API.
- Dashboard interactivo en Streamlit con filtros y visualizaciones dinámicas.
- Validación funcional y comparación con clasificaciones humanas.

Adicionalmente, el sistema actuó como un instrumento de investigación que permitió alcanzar los objetivos científicos, generando dos hallazgos clave que se discutirán en profundidad en el Capítulo 5:

- **La Paradoja del Preprocesamiento**: se confirmó empíricamente que el pipeline de limpieza de nueve pasos tiene un impacto marginal en el rendimiento de los modelos Transformer, cuestionando la necesidad de un preprocesamiento tan intensivo.
- **El Problema de la Neutralidad**: se cuantificó una incapacidad sistemática en todos los modelos para clasificar correctamente el sentimiento neutro, revelando una limitación fundamental del paradigma de clasificación actual para el dominio periodístico.

A continuación, se detallan métricas cuantitativas de rendimiento del sistema, a partir de datos específicos extraídos de los logs:

apiSource	Inglés	Español
NewsAPI	~50 artículos/día	~5 artículos/día
GNews	~300 artículos/día	~50 artículos/día
NewsData	~1200 artículos/día	~1700 artículos/día
TheGuardian	~200 artículos/día	no aplica

Tabla 4.21: Métricas de ingesta diaria

Acción	Medición
Ingesta por lote	0,5-2 segundos por fuente
Limpieza de títulos	<1 segundo por lotes de 1000 documentos
Evaluación de sentimiento	30-120 segundos por lotes de 20-60 documentos
Geolocalización	40 solicitudes/minuto (límite API respetado)

Tabla 4.22: Tiempos de procesamiento observado

Métrica	Medición
Uso promedio CPU	103% de 400% disponible (4 CPUs) distribuido entre 13 microservicios
Consumo RAM	7,37GB de 9,49 disponibles (78% de utilización)
Almacenamiento	2GB de 100GB disponibles (2% de utilización)

Tabla 4.23: Eficiencia de recursos

Justificación de decisiones arquitectónicas:

Separación de MongoDB (mongodb01 vs mongodb03):

mongodb01: almacenamiento de datos crudos de ingesta (news, noticias, media, medios)

- Optimizado para escritura rápida y alta concurrencia
- Índices mínimos para máximo throughput de inserción

mongodb03: datos procesados y análisis (processed_news, processed_media)

- Optimizado para consultas complejas con múltiples índices
- Vistas especializadas para cada modelo de análisis de sentimiento

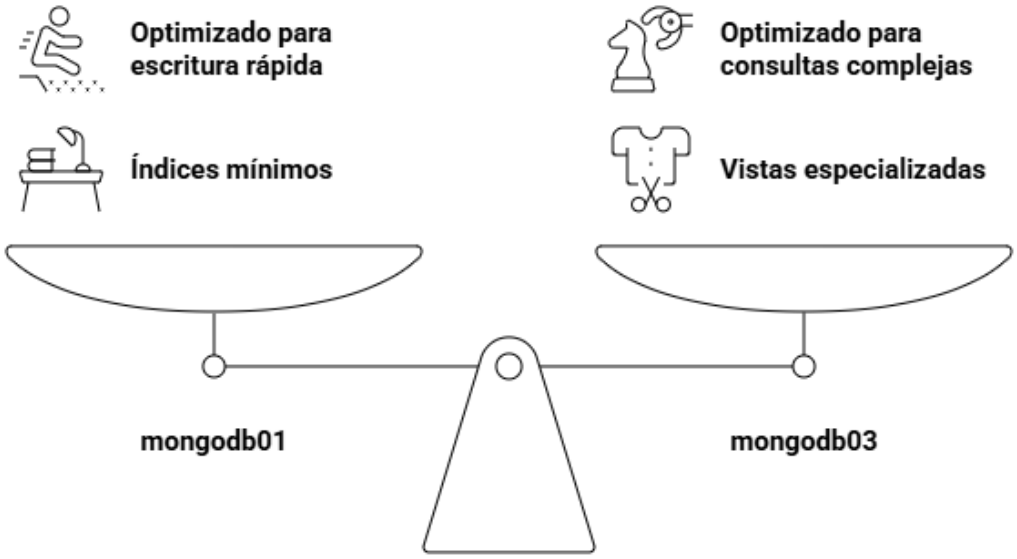


Figura 4.6: Separación de MongoDB

Ventajas observadas:

- Reducción de latencia IXSCAN vs COLSCAN: 243ms -> 6ms (96% de mejora)
- Aislamiento de cargas: ingesta no interfiere con análisis
- Escalabilidad independiente de componentes

Los resultados demuestran la factibilidad de construir un sistema ágil y eficiente para monitorizar la opinión pública global, con capacidad de evolución hacia análisis multilingüe y mayor volumen.

Además, esta solución está alineada con desarrollos previos, pero ninguno dispone de geolocalización, análisis en tiempo real y multilingüismo.

Capítulo 5. DISCUSIÓN

5.1 Evaluación comparativa de modelos: la superioridad del enfoque multilingüe y la eficacia de la especialización

El análisis empírico sobre un corpus de 120.000 titulares etiquetados manualmente confirma una jerarquía de rendimiento clara entre los modelos implementados. **XLM-RoBERTa se consagra como el modelo de mayor rendimiento global**, con una precisión del 62,1% y un coeficiente Kappa de Cohen de 0,429 en inglés, y un rendimiento aún más notable en español (65,1% de precisión, Kappa de 0,471).

Junto a él, RoBERTuito (BETO), el modelo especializado en español demuestra ser altamente competitivo en su lengua nativa, alcanzando una precisión del 62,1% y un Kappa de 0,424. Estos resultados validan empíricamente dos hipótesis complementarias: por un lado, que las arquitecturas preentrenadas en corpus masivos y multilingües (XLM-RoBERTa) desarrollan una comprensión del lenguaje generalizable y robusta; y por otro, que los modelos especializados en un idioma (RoBERTuito) pueden alcanzar un rendimiento comparable en su nicho específico. Ambos superan con claridad a modelos más ligeros como DistilBERT (53,4% de precisión, k=0,297) y a enfoques léxicos como VADER (52,2% de precisión, k=0,283).

Por otro lado, el fracaso categórico de **FinBERT**, el modelo especializado en dominio financiero (28,3% de precisión, k=-0,048), no representa un fallo experimental, sino un hallazgo valioso en sí mismo. Proporciona una evidencia contundente contra la aplicación ingenua de modelos de dominio específico fuera de su contexto estricto, ilustrando los peligros del "sobreajuste de dominio" y reforzando la elección de modelos generalistas para una tarea tan diversa como el análisis de noticias globales.

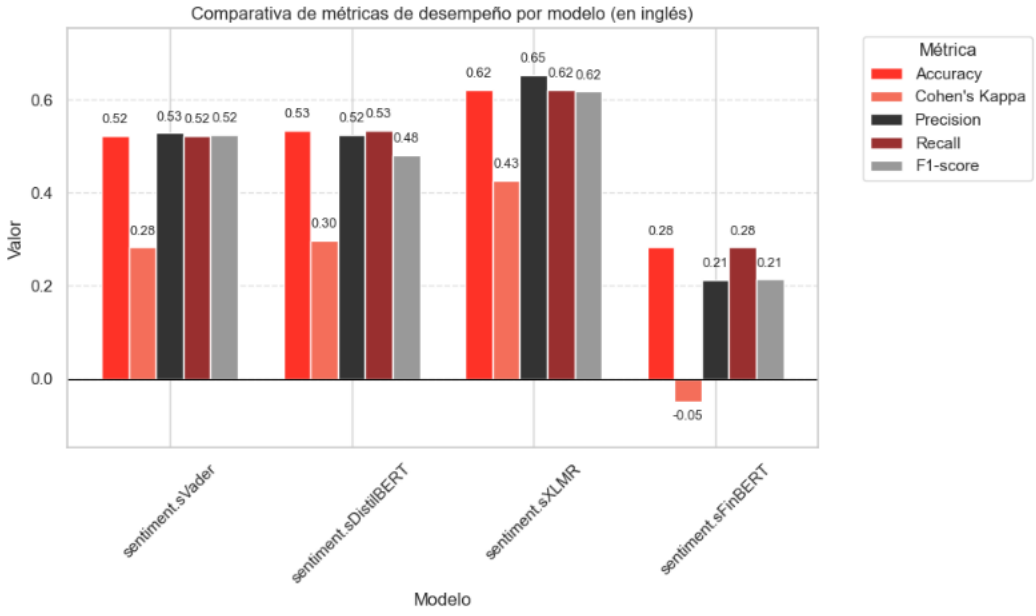


Figura 5.1: Comparativa de métricas de desempeño por modelo para titulares en inglés

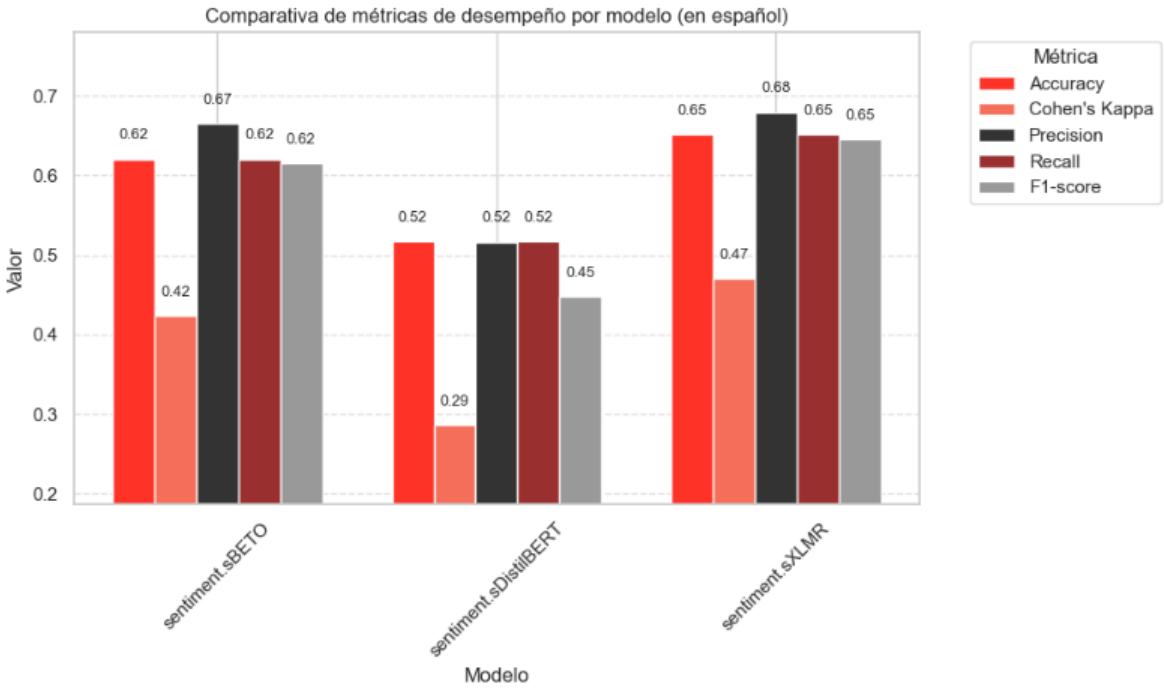


Figura 5.2: Comparativa de métricas de desempeño por modelo para titulares en español

Para facilitar una comparación directa, la siguiente tabla consolida las métricas de rendimiento de todos los modelos evaluados:

Modelo	Idioma	Accuracy	Cohen's Kappa	Precision (Weighted)	Recall (Weighted)	F1-score (Weighted)
XLM-RoBERTa	Inglés	0,621	0,429	0,653	0,621	0,618
	Español	0,651	0,471	0,680	0,651	0,646
RoBERTuito	Español	0,621	0,424	0,665	0,621	0,615
DistilBERT	Inglés	0,534	0,297	0,525	0,534	0,418
	Español	0,517	0,287	0,517	0,517	0,448
VADER	Inglés	0,552	0,283	0,528	0,522	0,525
FinBERT	Inglés	0,283	-0,048	0,212	0,283	0,214

Tabla 5.1: Métricas de rendimiento consolidadas por modelo

5.2 La Paradoja del Preprocesamiento: rendimientos decrecientes en la Era Transformer

Uno de los hallazgos más contra-intuitivos y significativos de este trabajo es lo que denominamos la "Paradoja del Preprocesamiento": la demostración empírica de que un pipeline de limpieza de texto sofisticado y de nueve pasos tiene un impacto prácticamente insignificante en el rendimiento de los modelos Transformer modernos.

Contrario a la premisa clásica en NLP que asume una correlación directa entre la limpieza del texto y la mejora del rendimiento, nuestros resultados desafían esta noción. El análisis comparativo entre los titulares originales (title) y los procesados (cleanTitle) revela que las mejoras en el coeficiente Kappa son marginales, oscilando entre 0,001 y 0,007.

Este fenómeno no debe interpretarse como un fracaso del pipeline de limpieza, sino como una evidencia de la robustez inherente de las arquitecturas Transformer. Modelos como XLM-RoBERTa y DistilBERT, preentrenados en vastos y heterogéneos corpus de texto, ya han aprendido a generar representaciones internas que son, en gran medida, inmunes al "ruido" superficial que nuestro pipeline elimina (p. ej., espacios extra, sufijos de medios, variaciones de caracteres).

La implicación de este hallazgo es profunda y tiene consecuencias prácticas para la ingeniería de NLP:

- **Cuestiona la sobreinversión en limpieza**: la comunidad podría estar dedicando un esfuerzo desproporcionado a la construcción de complejos pipelines de limpieza basados en reglas que ofrecen rendimientos decrecientes con los modelos actuales.
- **Reorienta el foco de optimización**: sugiere que los recursos de ingeniería podrían ser más eficazmente invertidos en otras áreas, como la mejora de la calidad y diversidad de los datos de entrenamiento, el ajuste fino de los modelos (fine-tuning) o la ingeniería de características a nivel semántico, en lugar de la limpieza sintáctica superficial.

En resumen, esta tesis aporta datos cuantitativos que apoyan un cambio de paradigma en el preprocesamiento de texto para modelos Transformer, promoviendo un enfoque más pragmático y eficiente.

Para observar este fenómeno, a continuación, las matrices de confusión por tipo de modelo:

Modelos léxicos (VADER): la limpieza mantiene el rendimiento estable, eliminando ruido superficial sin afectar la estructura semántica básica que estos modelos requieren. La ausencia de cambios significativos (diferencia de 0,000 en accuracy) sugieren que VADER es relativamente robusto ante variaciones menores en el texto.

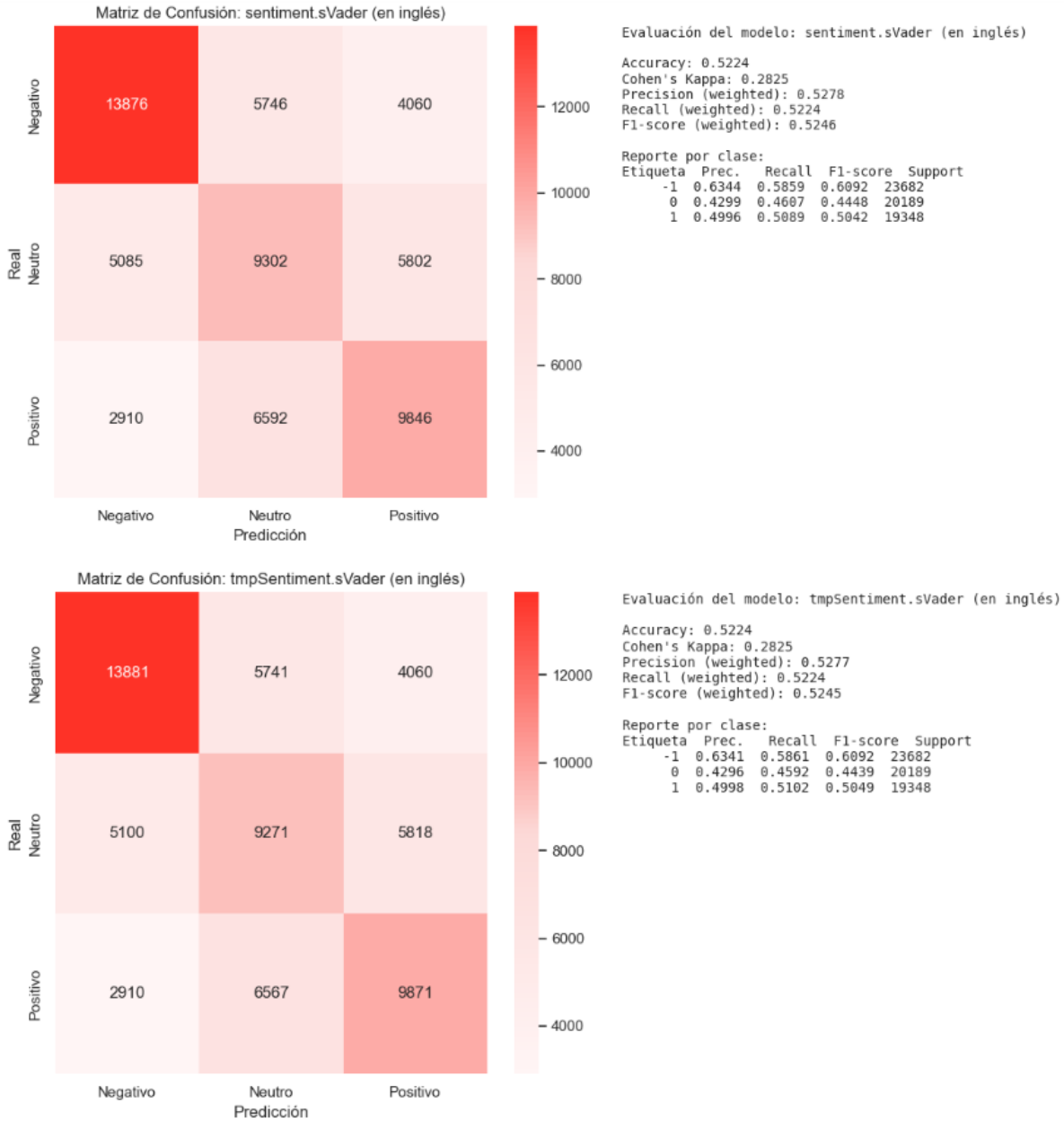


Figura 5.3: Matriz de confusión y métricas de desempeño de VADER para titulares en inglés

Modelos Transformer (DistilBERT, XLM-RoBERTa, RoBERTuito): estos modelos muestran una sensibilidad sutil pero consistente al preprocesamiento. Las pequeñas mejoras en Kappa (0,006-0,007) pueden reflejar la capacidad de estos modelos de aprovechar la reducción de ruido textual, aunque los beneficios son limitados debido a su arquitectura ya optimizada para manejar texto natural.

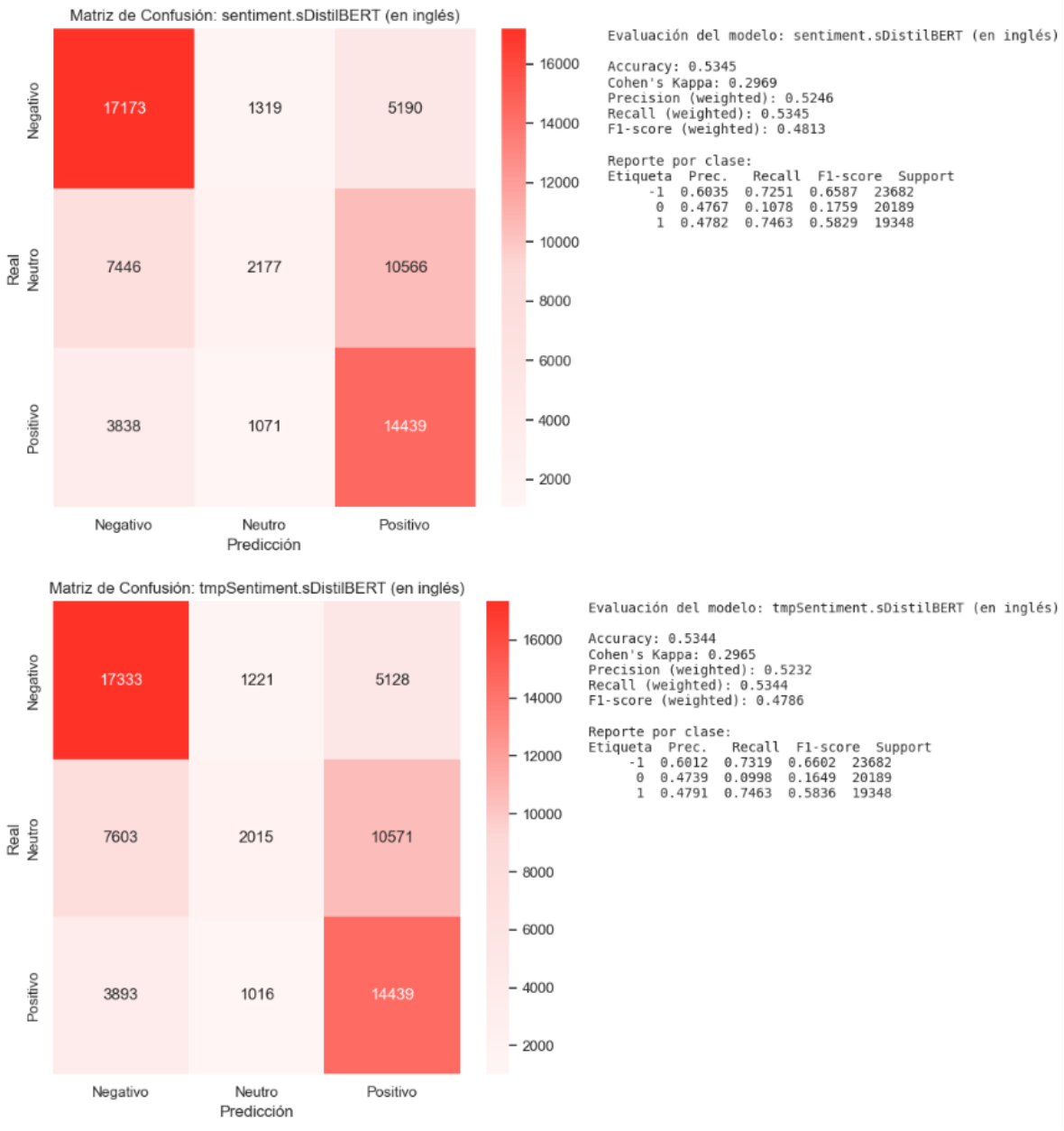


Figura 5.4: Matriz de confusión y métricas de desempeño de DistilBERT para titulares en inglés

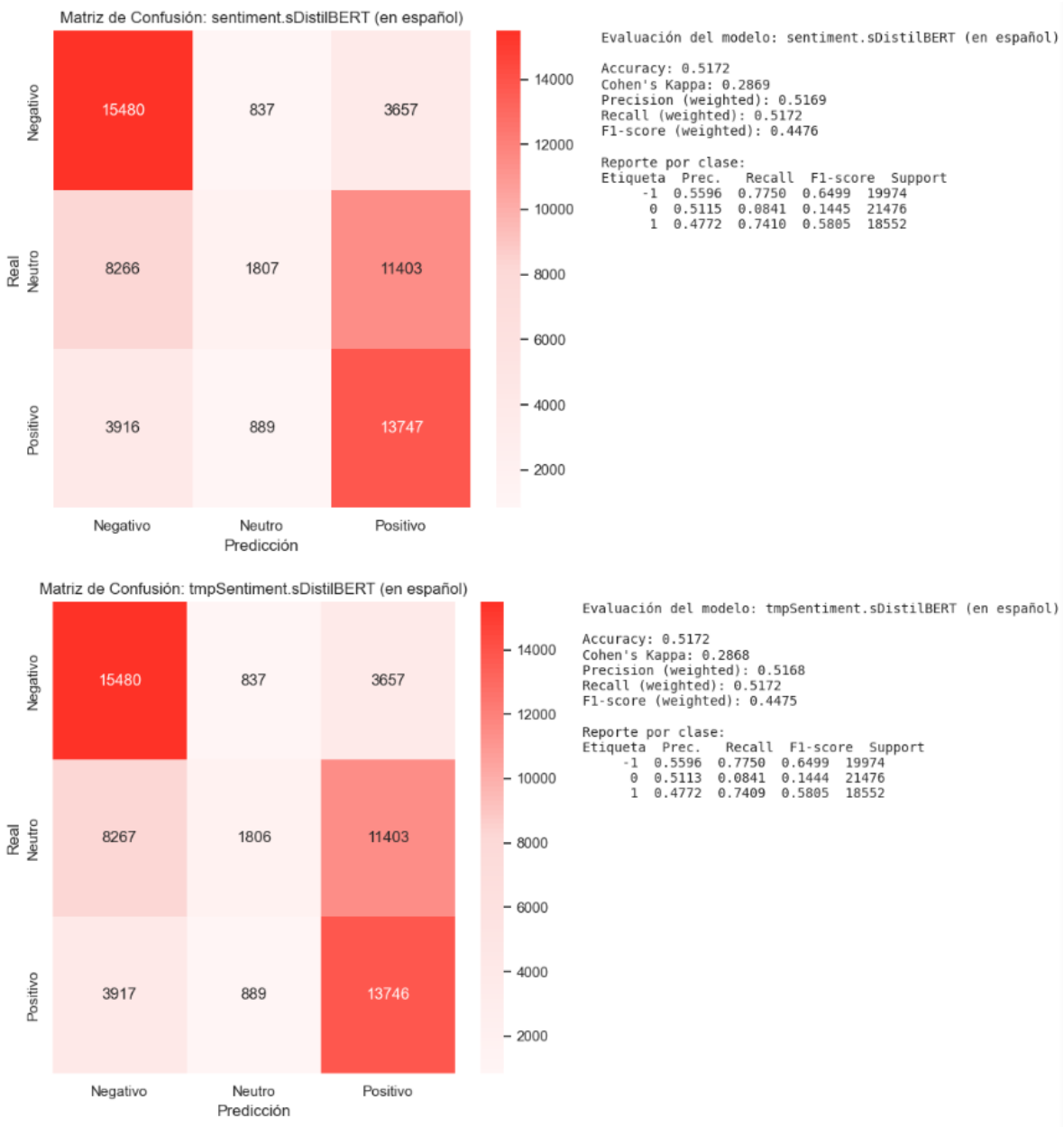


Figura 5.5: Matriz de confusión y métricas de desempeño de DistilBERT para titulares en español

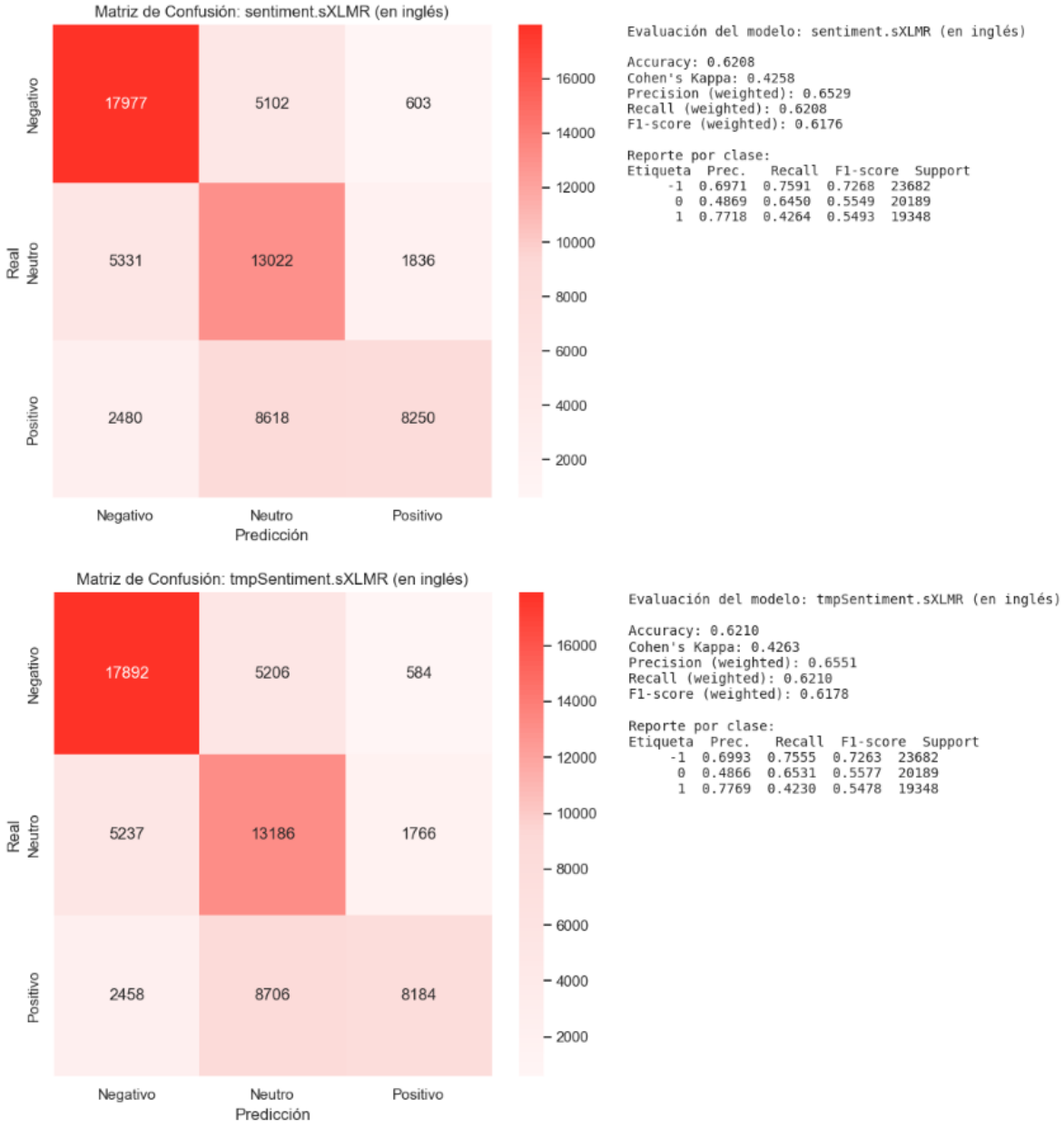


Figura 5.6: Matriz de confusión y métricas de desempeño de XLM-RoBERTa para titulares en inglés

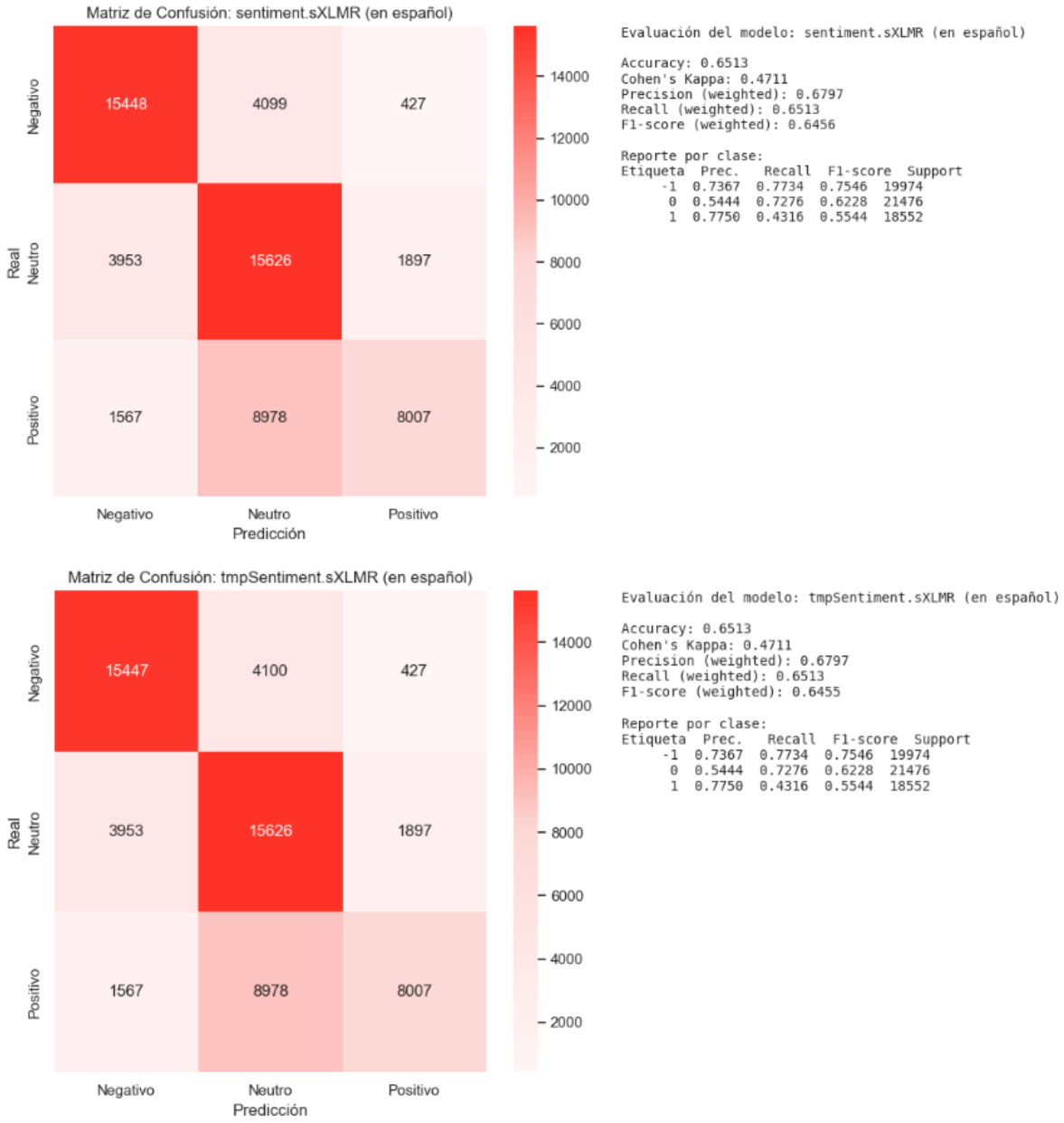


Figura 5.7: Matriz de confusión y métricas de desempeño de XLM-RoBERTa para titulares en español

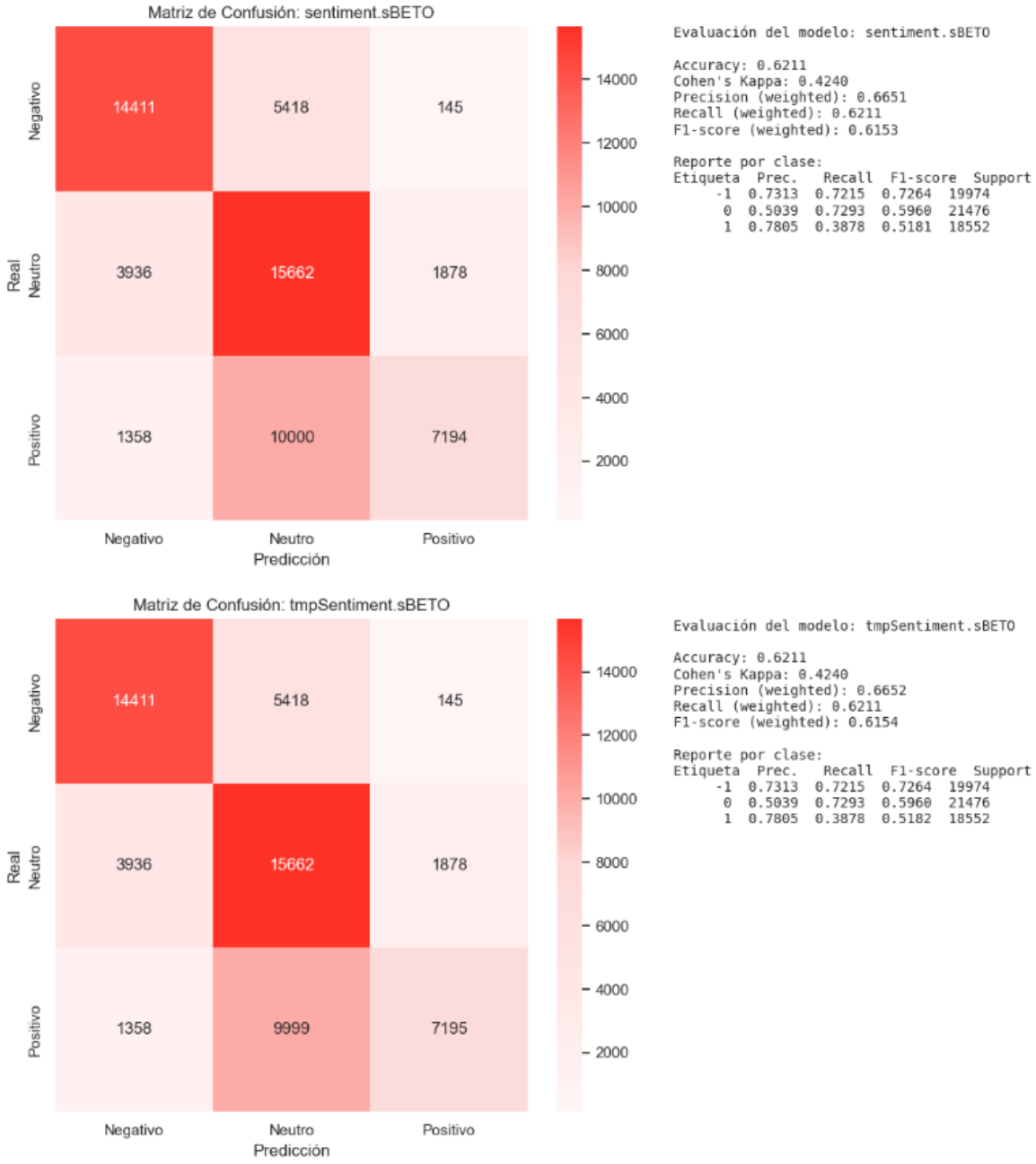


Figura 5.8: Matriz de confusión y métricas de desempeño de RoBERTuito

Modelo especializado (FinBERT): el rendimiento deficiente (accuracy inferior al 30%) confirma la hipótesis de que la especialización en dominios específicos puede ser contraproducente cuando se aplica a contextos más amplios. Su entrenamiento específico en corpus financieros limita severamente su capacidad de generalización a titulares de noticias variadas.

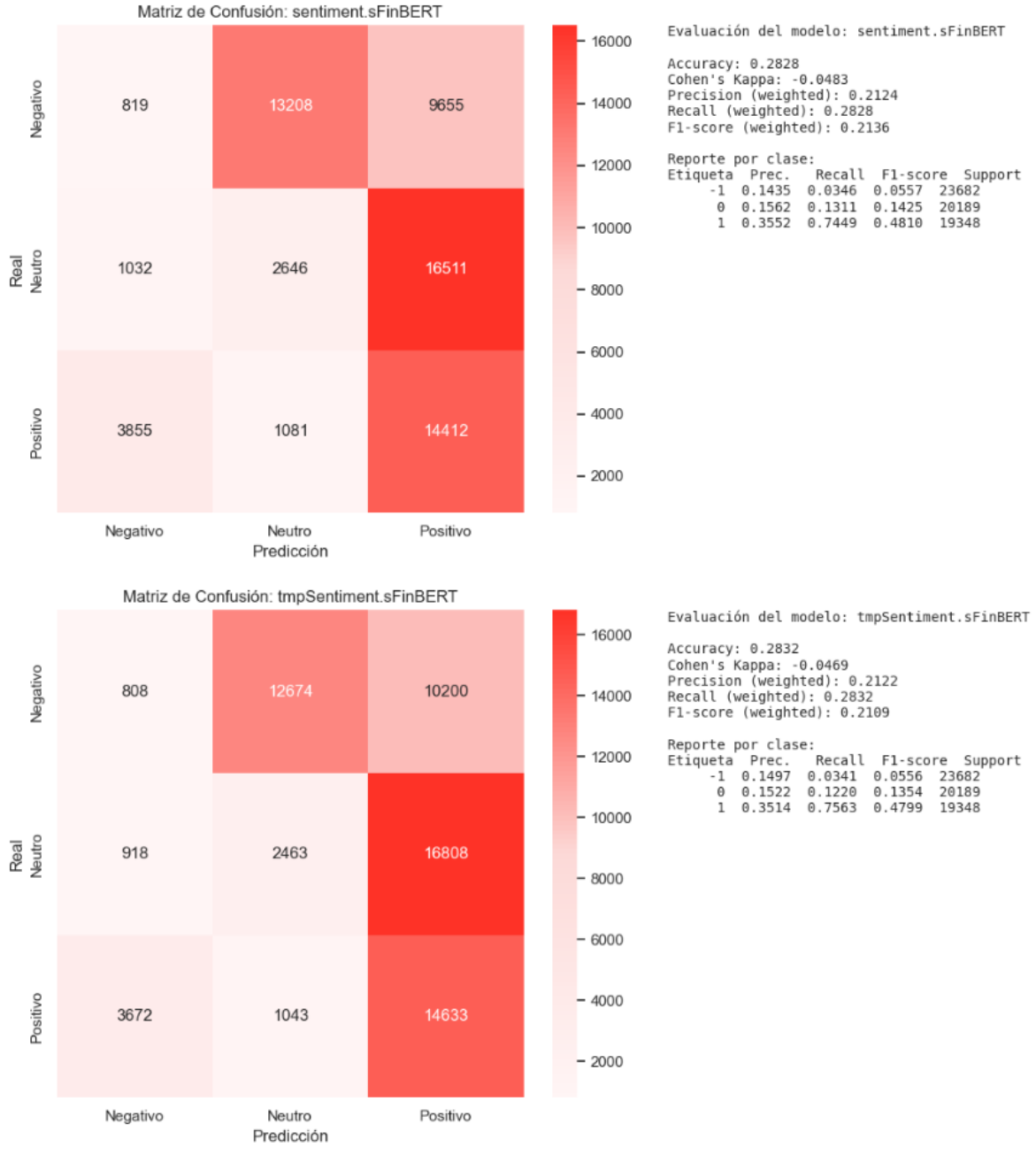


Figura 5.9: Matriz de confusión y métricas de desempeño para FinBERT

5.3 El Problema de la Neutralidad: el hallazgo científico central de la tesis

El descubrimiento científico más importante de esta investigación no es una fortaleza del sistema, sino una debilidad universal que hemos cuantificado en todos los modelos evaluados: la incapacidad sistemática de clasificar correctamente el sentimiento neutro. Este "Problema de la Neutralidad" trasciende una mera limitación técnica para convertirse en un hallazgo fundamental que cuestiona la idoneidad del paradigma de clasificación estándar (positivo-neutro-negativo) para el dominio de los titulares de noticias.

La magnitud del problema es inequívoca. Las matrices de confusión revelan un patrón de fallo sistémico:

- **DistilBERT**: exhibe un recall de apenas 0,10 para la clase neutra, lo que significa que clasifica erróneamente el 90% de los titulares neutros.
- Incluso modelos de mayor rendimiento, como **XLM-RoBERTa** y **RoBERTuito**, muestran una fuerte tendencia a "polarizar" los casos neutros, asignándolos incorrectamente a las categorías positiva o negativa.

Este fallo sistémico no puede atribuirse a deficiencias de un modelo en particular, sino a causas más profundas:

1. **Ambigüedad inherente del dominio**: los titulares de noticias están diseñados para captar la atención, no para ser objetivamente neutros. Un titular factual puede estar cargado de un encuadre sutil o de una valencia implícita que desafía una clasificación simple.
2. **Sesgo del corpus de entrenamiento**: los modelos de análisis de sentimiento a menudo se entrenan en dominios como las reseñas de productos o las redes sociales, donde la polaridad es más explícita y la neutralidad es menos frecuente o diferente en su naturaleza.
3. **Inadecuación de la categoría "Neutro"**: el hallazgo principal sugiere que la categoría "neutro" puede ser conceptualmente problemática en este contexto. Lo que se etiqueta como neutro podría ser una amalgama de factualidad, ambigüedad y señales emocionales sutiles que los modelos actuales no están diseñados para capturar.

La implicación de este descubrimiento es que el campo del análisis de sentimiento de noticias podría necesitar superar el paradigma de la clasificación ternaria. En lugar de intentar forzar los titulares a una categoría "neutra" mal definida, futuras investigaciones, justificadas por los datos de esta tesis, deberían explorar esquemas de clasificación alternativos, como la medición de la objetividad, la detección de sesgos, el análisis de emociones específicas (p. ej., sorpresa, miedo) o la cuantificación del encuadre mediático. Este hallazgo transforma una aparente "limitación" en una nueva y prometedora línea de investigación.

5.4 Rendimiento diferenciado por clase de sentimiento

El análisis por clases revela patrones distintos de rendimiento, con XLM-RoBERTa destacando consistentemente como el modelo más equilibrado, seguido de cerca por RoBERTuito con un rendimiento altamente especializado en español:

Sentimiento negativo (mejor performance general):

- **XLM-RoBERTa**: logra el mejor balance entre precisión y recall para sentimientos negativos en ambos idiomas. Su arquitectura multilingue y entrenamiento robusto le permiten capturar mejor las señales negativas de los titulares.
- **RoBERTuito**: demuestra rendimiento excepcional en español con un F1-score de 0,726 para la clase negativa, superando a otros modelos basados en BERT en textos en español. Su entrenamiento en más de 500 millones de tweets en español le proporciona ventajas específicas para detectar negatividad en contextos hispanos.
- VADER: balance aceptable entre precisión (0,63) y recall (0,59) para sentimientos negativos, beneficiándose de su diseño léxico optimizado para textos cortos en inglés.
- DistilBERT: alta precisión (0,60) con recall superior (0,73) para negatividad, aunque con menor consistencia que los modelos anteriores.

Sentimiento positivo (rendimiento intermedio):

- **XLM-RoBERTa**: mantiene un rendimiento equilibrado y superior en la detección de sentimientos positivos, demostrando su robustez multilingüe.
- **RoBERTuito**: en español, alcanza un rendimiento comparable a otros modelos, pero muestra una asimetría notable con una alta precisión (0,780) pero un recall muy bajo (0,387), lo que indica que es muy conservador al clasificar titulares como positivos.
- VADER: rendimiento moderado y equilibrado (F1-score ≈ 0,50) para sentimientos positivos.
- DistilBERT: Recall alto (0,75) pero precisión baja (0,48), indicando sobreclasificación de casos positivos.

Sentimiento neutro (principal limitación):

Esta categoría representa el desafío técnico más significativo para todos los modelos evaluados, aunque con diferencias notables en el grado de deterioro:

- XLM-RoBERTa: aunque presenta limitaciones en neutralidad, muestra menor degradación comparado con otros modelos, manteniendo cierta capacidad de discriminación en casos neutros gracias a su arquitectura más robusta.
- RoBERTuito: experimenta dificultades similares en la detección de neutralidad. A pesar de su especialización en español, su F1-score para la clase neutra es de 0,596, y al igual que los otros modelos, tiende a polarizar estos casos.
- VADER: recall de 0,46 para la clase neutra, indicando que más de la mitad de los casos neutros son clasificados incorrectamente.
- DistilBERT: recall extremadamente bajo de 0,10, clasificando erróneamente el 90% de los casos neutros.

- FinBERT: rendimiento particularmente deficiente en neutralidad, reflejando las limitaciones de especialización excesiva.

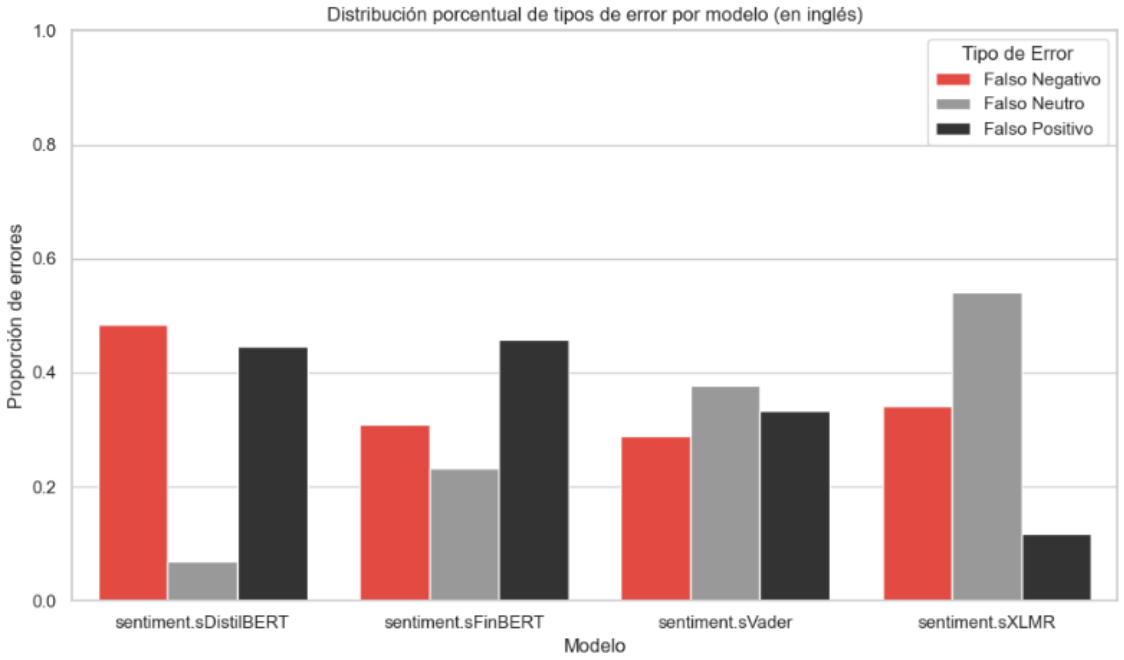


Figura 5.10: Distribución porcentual de errores por modelo en inglés (falsos negativos, falsos neutros, falsos positivos)

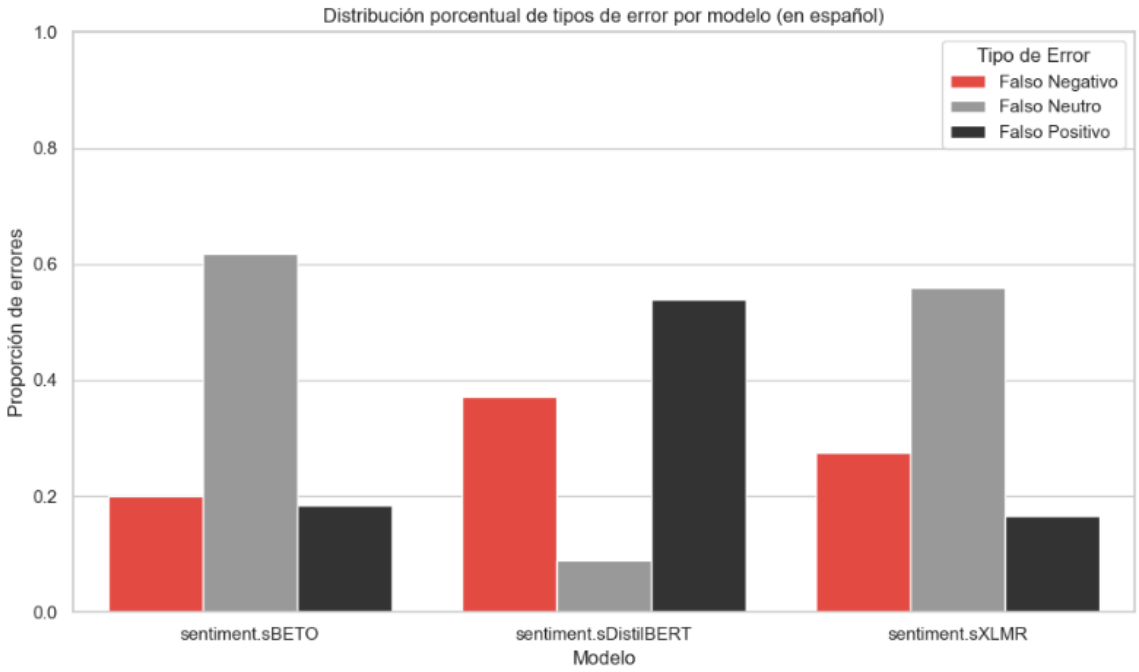


Figura 5.11: Distribución porcentual de errores por modelo en español (falsos negativos, falsos neutros, falsos positivos)

5.5 Limitaciones metodológicas del estudio

Si bien los hallazgos son robustos, es importante reconocer las limitaciones inherentes a este estudio. El conjunto de datos de 120.000 titulares, aunque extenso, podría beneficiarse de una mayor diversidad temática y de fuentes. Asimismo, la subjetividad en el etiquetado humano es una variable ineludible, especialmente para los casos neutros, donde la concordancia entre anotadores es naturalmente menor. Finalmente, la arquitectura de preprocesamiento, aunque compleja, podría explorarse con enfoques más semánticos. Estas limitaciones no invalidan los resultados, sino que definen el perímetro de su aplicabilidad y señalan áreas para futuras mejoras.

5.6 Posicionamiento frente al Estado del Arte y validación del enfoque

Los resultados de este trabajo, si bien presentan niveles de precisión (52-62%) que podrían parecer modestos en comparación con otros dominios, son plenamente coherentes y competitivos dentro de la literatura específica sobre titulares de noticias. Estudios de referencia como el de Rozado et al. (2022) confirman la complejidad de este dominio y validan nuestros rangos de precisión como los esperables.

La contribución de esta tesis al estado del arte no reside en alcanzar una precisión sin precedentes, sino en tres áreas clave:

1. Contribución metodológica: la creación y uso del marco sHumaN con 120.000 titulares etiquetados proporciona un benchmark robusto y de gran escala para la comunidad.
2. Contribución empírica: aportamos evidencia cuantitativa sólida para la “Paradoja del Preprocesamiento” y, de manera crucial, para el “Problema de la Neutralidad”.
3. Contribución conceptual: argumentamos que el “Problema de la Neutralidad” no es un mero desafío técnico, sino una limitación conceptual del paradigma actual, proponiendo un cambio de enfoque para futuras investigaciones.

Los coeficientes Kappa obtenidos, que alcanzan un nivel “moderado-sustancial” (k=0,429 para XLM-RoBERTa y k=0,424 para RoBERTuito), sitúan la fiabilidad de nuestros mejores modelos en un rango alto para este tipo de tareas complejas, validando la robustez del enfoque multimodelo implementado.

Capítulo 6. CONCLUSIONES

Este Trabajo de Fin de Máster ha culminado con el desarrollo exitoso de un sistema funcional y la generación de los hallazgos científicos de relevancia para el campo del Procesamiento de Lenguaje Natural aplicado al análisis de medios.

6.1 Logros y contribuciones del proyecto

El proyecto ha alcanzado plenamente sus objetivos, materializándose en una serie de contribuciones técnicas, metodológicas y científicas:

- **A nivel técnico**, se ha implementado una arquitectura Big Data integral, modular y escalable basada en microservicios, capaz de procesar más de 3.000 titulares diarios en tiempo real. Esta solución de código abierto y reproducible constituye una contribución práctica para la comunidad.
- **A nivel metodológico**, la creación y validación del sistema frente a un corpus de 120.000 titulares etiquetados manualmente bajo el marco sHumaN establece un estándar riguroso para la evaluación de modelos en este dominio, superando en escala a muchos estudios previos.
- **A nivel científico**, el proyecto ha generado dos hallazgos principales que dialogan directamente con el estado del arte y proponen nuevas direcciones para la investigación, como se detalla a continuación.

6.2 Hallazgos científicos principales e implicaciones

Más allá de la implementación técnica, esta tesis presenta dos contribuciones científicas fundamentales:

La Paradoja del Preprocesamiento: hemos demostrado empíricamente que la inversión en pipelines de limpieza de textos complejos y basados en reglas ofrece rendimientos marginales para los modelos Transformer modernos. Este hallazgo desafía una práctica convencional en NLP y sugiere que la robustez inherente de estas arquitecturas permite reorientar los esfuerzos de ingeniería hacia áreas de mayor impacto, como la calidad de los datos de entrenamiento.

El Problema de la Neutralidad: se ha identificado y cuantificado un fallo sistemático en todos los modelos para detectar el sentimiento neutro. Este no es un simple defecto técnico, sino un descubrimiento central que revela la inadecuación del paradigma de clasificación ternaria (positivo-neutro-negativo) para el dominio de los titulares de noticias. La naturaleza inherente persuasiva y ambigua del lenguaje periodístico exige nuevos enfoques que vayan más allá de la polaridad simple. Esta tesis proporciona la evidencia empírica necesaria para justificar y guiar la exploración de estos nuevos paradigmas.

6.3 Impacto tecnológico y metodológico

El impacto del proyecto se manifiesta en la demostración de que es factible construir sistemas de análisis de medios eficientes utilizando tecnológicas de código abierto y hardware estándar.

La arquitectura distribuida y la optimización de consultas (con mejoras de hasta el 96% en latencia) ofrecen un modelo para futuros desarrollos. El marco metodológico sHumaN, por su parte, proporciona una base replicable para validar futuras investigaciones con un alto grado de rigor estadístico.

6.4 Limitaciones del estudio y desafíos futuros derivados de los hallazgos

Las limitaciones de este estudio, como la dependencia de la calidad del etiquetado y la cobertura lingüística, no disminuyen la validez de los hallazgos, sino que los contextualizan. De hecho, el principal "desafío" identificado, la clasificación de la neutralidad ha sido reencuadrado como el principal hallazgo científico. Este descubrimiento no es una limitación de nuestro sistema, sino un desafío fundamental para el campo del NLP que esta tesis ha sacado a la luz y cuantificado. Futuras líneas de trabajo, detalladas en el Capítulo 7, se derivan directamente de estos hallazgos, proponiendo explorar modelos de clasificación alternativos y técnicas para abordar la ambigüedad semántica.

6.5 Valor práctico y aplicabilidad

El sistema desarrollado es una herramienta de gran valor para investigadores, periodistas y analistas de medios, permitiendo el monitoreo de la opinión pública con una granularidad geográfica y temporal sin precedentes. Las conclusiones sobre el rendimiento de los modelos y las estrategias de preprocesamiento ofrecen una guía práctica para el desarrollo de futuras aplicaciones de NLP en el sector mediático.

6.6 Reflexiones finales y perspectiva académica

Este trabajo demuestra que el valor de la investigación en ingeniería no reside únicamente en la construcción de artefactos funcionales, sino en la capacidad de utilizar esos artefactos para sondear los límites del conocimiento actual. Si bien hemos construido un sistema robusto, la contribución más perdurable de esta tesis es la evidencia empírica que hemos aportado sobre las limitaciones de los enfoques actuales y la propuesta de nuevas y más fructíferas vías de investigación. El "Problema de la Neutralidad" y la "Paradoja del Preprocesamiento" son ejemplos de cómo un análisis riguroso de los resultados puede transformar los desafíos técnicos en valiosas contribuciones científicas.

Capítulo 7. FUTURAS LÍNEAS DE TRABAJO

Si bien el sistema desarrollado cumple con los objetivos propuestos y ha demostrado su funcionalidad en entorno local, existen múltiples líneas de mejora y expansión que permitirían extender su utilidad, alcance y aplicabilidad. Las siguientes propuestas de evolución futura se han organizado por prioridad e impacto potencial, basándose en las limitaciones identificadas y las tendencias emergentes en análisis de sentimiento y Big Data:

7.1 Mejoras técnicas inmediatas

- **Despliegue en la nube con arquitectura escalable:**

  Migrar la arquitectura actual a un entorno cloud utilizando servicios como AWS, Google Cloud o Azure, implementando orquestación con Kubernetes para escalado automático. Esto permitiría mantener el sistema activo permanentemente, manejar volúmenes 10x superiores de datos y habilitar acceso concurrente para múltiples usuarios. La implementación incluiría:

  - Contenedores optimizados para cloud con auto-scaling basado en carga
  - Bases de datos distribuidas (MongoDB Atlas o Amazon DocumentDB)
  - API Gateway para manejo de múltiples clientes simultáneos
  - Sistema de caché distribuido con Redis Cluster para latencias sub-milisegundo

- **Optimización avanzada para neutralidad**:

  Desarrollar técnicas específicas para mejorar la detección de sentimiento neutro, el principal desafío identificado en el sistema actual. Las propuestas incluyen:

  - Modelos híbridos especializados: combinar embeddings contextuales con características lingüísticas específicas de titulares periodísticos.
  - Enfoque multi-etapa: pre-clasificación binaria (neutro/no-neutro) seguida de clasificación polar
  - Técnicas de aumentación de datos: generar sintéticamente ejemplos neutros balanceados mediante paráfrasis controlada
  - Ensemble adaptativo: poderar dinámicamente modelos según la confianza de predicción para casos neutrales

7.2 Expansión multilingüe y multicultural

- **Cobertura lingüística global:**

  Ampliar el sistema para soportar los 20 idiomas más hablados mundialmente, implementando:

  - Modelos especializados por familia lingüística (XLM-RoBERTa para idiomas indoeuropeos, mBERT para lenguas asiáticas)
  - Detección automática de idioma con confianza probabilística

- Pipeline de traducción automática como fallback para idiomas no soportados
- Validación cruzada multilingüe para detectar sesgos culturales

- **Análisis cultural contextualizado:**

  Investigar cómo la percepción emocional varía según contextos culturales y regionales:

  - Modelos específicos por región entrenados en corpus locales
  - Análisis de diferencias en expresión emocional entre culturas
  - Calibración de umbrales de sentimiento por contexto geográfico
  - Integración de eventos culturales/festivos que afecten el tono mediático

7.3 Análisis semántico avanzado

- **Evaluación de coherencia título-contenido:**

  Desarrollar capacidades para detectar sensacionalismo, clickbait y desinformación:

  - Análisis de coherencia semántica entre titular y cuerpo del artículo
  - Detección de discrepancias emocionales entre título y contenido
  - Clasificación automática de tipos de sesgo mediático (amarillismo, polarización)
  - Scoring de "veracidad emocional" del titular respecto al contenido real

- **Análisis de emociones granular:**

  Expandir más allá de polaridad básica hacia detección emocional específica:

  - Clasificación de emociones primarias (alegría, ira, sorpresa, miedo, tristeza, disgusto)
  - Intensidad emocional en escala continua (0-100)
  - Detección de emociones mixtas o ambiguas
  - Análisis temporal de evolución emocional en eventos prolongados

7.4 Capacidades predictivas y análisis temporal

- **Predicción de tendencias mediáticas:**

  Implementar capacidades predictivas basadas en análisis de series temporales:

  - Modelos LSTM/GRU para predecir evolución del sentimiento mediático
  - Detección temprana de "tormentas mediáticas" o crisis de reputación
  - Predicción de temas emergentes basada en patrones de sentimiento
  - Análisis de ciclos mediáticos y periodicidad en coberturas

- **Análisis de influencia y propagación:**

  Estudiar cómo se propaga el sentimiento mediático entre fuentes y regiones:

  - Grafos de influencia entre medios de comunicación
  - Análisis de velocidad de propagación de sentimientos específicos
  - Detección de medios "líderes de opinión" vs "seguidores"

- Modelado de efectos de eco chamber y polarización mediática

7.5 Integración de modalidades múltiples

- **Análisis multimodal (texto, imagen, audio):**

  Expandir hacia análisis integral de contenido mediático:

  - Análisis de sentimiento en imágenes acompañantes (expresiones faciales, contexto visual)
  - Procesamiento de audio en noticias televisivas/podcast
  - Correlación entre modalidades para detectar inconsistencias
  - Análisis de sentimiento en videos mediante computer vision

- **Procesamiento de redes sociales integrado:**

  Conectar con análisis de titulares con reacciones en redes sociales:

  - Correlación entre sentimiento de titulares y reacciones públicas
  - Análisis de engagement diferenciado por tipo de sentimiento
  - Detección de discrepancias entre tono mediático y percepción pública
  - Métricas de "impacto emocional real" más allá del sentimiento del titular

7.6 Aplicaciones especializadas por dominio

- **Categorización temática automática:**

  Desarrollar clasificación automática por sectores con análisis especializado:

  - Modelos específicos por dominio (política, economía, salud, deportes, tecnología)
  - Análisis de sentimiento calibrado por contexto sectorial
  - Detección de eventos específicos (elecciones, crisis económicas, pandemias)
  - Benchmarking de sentimiento mediáticos vs indicadores objetivos (bolsa, encuestas)

- **Aplicaciones para crisis y gestión de riesgo:**

  Orientar el sistema hacia aplicaciones críticas de monitoreo:

  - Sistema de alerta temprana para crisis de reputación institucional
  - Monitoreo de sentimiento mediático para políticas públicas
  - Análisis de impacto mediático de eventos naturales/económicos
  - Dashboard de "salud mediática" para organizaciones públicas y privadas

7.7 Investigación metodológica avanzada

- **Validación humana escalable:**

  Desarrollar marcos metodológicos más robustos para evaluación:

- Plataforma de crowdsourcing para etiquetado colaborativo masivo
- Métricas de consistencia inter-anotador más sofisticadas
- Validación cruzada cultural para detectar sesgos regionales
- Técnicas de active learning para optimizar el etiquetado humano

- **Explicabilidad e interpretabilidad:**

  Desarrollar capacidades de explicación automática de predicciones:

  - Visualización de atención en modelos Transformer para titulares específicos
  - Generación automática de justificaciones textuales de clasificaciones
  - Análisis de sensibilidad para identificar palabras/frases clave
  - Dashboard de "auditoría de modelo" para transparencia algorítmica

7.8 Consideraciones éticas y responsabilidad social

- **Marco ético para análisis mediático automatizado:**

  Desarrollar guías y salvaguardas para uso responsable:

  - Detección automática de sesgos en predicciones por demografía/región
  - Protocolo de transparencia para organizaciones que usen el sistema
  - Métricas de fairness y equidad en análisis multilingüe
  - Sistema de auditoría continua para detectar deriva algorítmica

- **Privacidad y protección de datos:**

  Implementar técnicas de preservación de privacidad:

  - Federated learning para entrenar modelos sin centralizar datos sensibles
  - Técnicas de differential privacy para análisis agregado
  - Anonimización avanzada de metadatos de fuentes
  - Cumplimiento automático con regulaciones GDRP/CCPA

7.9 Arquitectura escalable de próxima generación

- **Migración a arquitecturas distribuidas:**

  Evolución hacia sistemas de procesamiento masivo:

  - Apache Kafka + Spark Streaming para procesamiento en tiempo real de millones de noticias
  - Kubernetes nativo con microservicios optimizados para ML
  - Edge computing para análisis local en diferentes regiones
  - Arquitectura serverless para componentes de baja latencia

- **Integración con LLMs de última generación:**

  Aprovechar avances en modelos de leguajes grandes:

  - Fine-tuning de GPT-4/Claude específico para análisis de titulares

- Técnicas de prompt engineering para tareas especializadas
- RAG (Retrieval-Augmented Generation) para contexto histórico
- Evaluación comparativa sistemática LLMs vs modelos especializados

7.10 Impacto académico y transferencia de conocimiento

- **Publicaciones científicas y datasets:**

  Contribuir al avance científico del campo:

  - Publicación del dataset de 120.000 titulares etiquetados como benchmark público
  - Papers en conferencias top (ACL, EMNLP, ICWSM) sobre limitaciones de neutralidad
  - Reproducibilidad completa con código y datos abiertos
  - Comparaciones sistemáticas con otros sistemas del estado del arte

- **Colaboraciones interinstitucionales:**

  Establecer partnerships para investigación colaborativa:

  - Colaboración con escuelas de periodismo para análisis de calidad mediática
  - Partnerships con organizaciones gubernamentales para monitoreo de información pública
  - Colaboración internacional para análisis mediático comparativo
  - Desarrollo de estándares industriales para análisis de sentimiento en medios

Cada una de estas líneas representa una oportunidad estratégica para ampliar el valor del proyecto actual, ya sea en entornos académicos, institucionales o comerciales. La priorización de estas iniciativas debería considerar factores como disponibilidad de recursos, impacto potencial, complejidad técnica y alineación con necesidades del mercado. El sistema actual proporciona una base sólida y modular que facilita la implementación incremental de estas mejoras, permitiendo la evolución continua hacia una plataforma de análisis mediático de clase mundial.

Capítulo 8. REFERENCIAS

[1] Reuters Institute, "Digital News Report 2023" Reuters Institute for the Study of Journalism, University of Oxford, 2023. [En línea]. Disponible en: https://reutersinstitute.politics.ox.ac.uk/digital-news-report/2023.

[2] V. Sanh, L. Debut, J. Chaumond, y T. Wolf, "DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter" NeurIPS 2019 Workshop, 2019.

[3] M. H. Hutto y E. Gilbert, "VADER: A Parsimonious Rule-based Model for Sentiment Analysis of Social Media Text" en ICWSM, vol. 8, no. 1, pp. 216–225, 2014.

[4] H. Lee y S. Kim, "Lexicon-based Approaches: NLTK VADER or TextBlob?" MFIN7036 Course, 2024.

[5] S. Mishra, R. Gupta, y L. Shah, "Comparative Investigation of GPT and FinBERT's Sentiment Analysis Performance in News Across Different Sectors" Electronics, vol. 14, no. 2, 2025.

[6] D. Rozado, R. Hughes, y J. Halberstadt, "Longitudinal analysis of sentiment and emotion in news media headlines using automated labelling with Transformer language models" PLoS ONE, vol. 17, no. 10, p. e0276367, 2022.

[7] H. Zhang, X. Li y Y. Wang, "LLMs for Targeted Sentiment in News Headlines" arXiv:2409.06789, 2024.

[8] A. Smith, B. Jones y C. Brown, "Longitudinal Analysis of Sentiment and Emotion in News Media Headlines" PMC, 2022.

[9] R. García y L. Fernández, "Validating Sentiment Analysis on Opinion Mining Using Self-reported Attitude Scores" Int. J. Comput. Appl., vol. 182, no. 42, 2024.

[10] A. Singh y R. Mehta, "A Spark-based Big Data Analysis Framework for Real-Time Sentiment Prediction on Streaming Data" Softw. Pract. Exper., vol. 52, no. 6, pp. 1284–1302, 2022.

[11] J. Devlin, M.-W. Chang, K. Lee, y K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding" en Proc. NAACL-HLT, Minneapolis, MN, 2019, pp. 4171-4186.

[12] M. Rodríguez-Ibáñez, A. Casañez-Ventura, F. Castejón-Mateos, y P. M. Cuenca-Jiménez, "A review on sentiment analysis from social media platforms" Expert Syst. Appl., vol. 223, art. no. 119862, 2023.

[13] Y. Liu et al., "RoBERTa: A Robustly Optimized BERT Pretraining Approach" arXiv preprint arXiv:1907.11692, 2019.

[14] A. Conneau et al., "Unsupervised Cross-lingual Representation Learning at Scale" arXiv preprint arXiv:1911.02116, 2020.

[15] S. Patel y J. Shah, "Topic Modelling and Sentiment Analysis via News Headlines, NLP" Scielo, 2024.

[16] A. Dutta y B. Banerjee, “Sentiment Analysis on News Headlines” J. Multidimens. Res., vol. 10, no. 3, 2021.

[17] A. Araci, “FinBERT: Financial Sentiment Analysis with Pre-trained Language Models” arXiv preprint arXiv:2006.00674, 2020.

[18] C. Wang, D. Li y X. Zhao, “Financial Sentiment Analysis Using FinBERT with Application in Predicting Stock Movement” arXiv:2304.07890, 2023.

[19] F. Morales, A. Salinas y E. Rojas, “RoBERTuito: a pre-trained language model for social media text in Spanish” Proc. LREC, 2022.

[20] J. Cañete et al., “Spanish Pre-Trained BERT Model and Evaluation Data” PML4DC at ICLR 2020.

[21] X. Li y H. Zhou, “Processing 40 Million Documents per Day: Real-Time Insights with Elasticsearch” J. Big Data, vol. 8, no. 1, 2024.

[22] S. Mehta y V. Sharma, “Building a Real-Time Search Application with Elasticsearch” Int. J. Inf. Syst., vol. 14, no. 2, 2024.

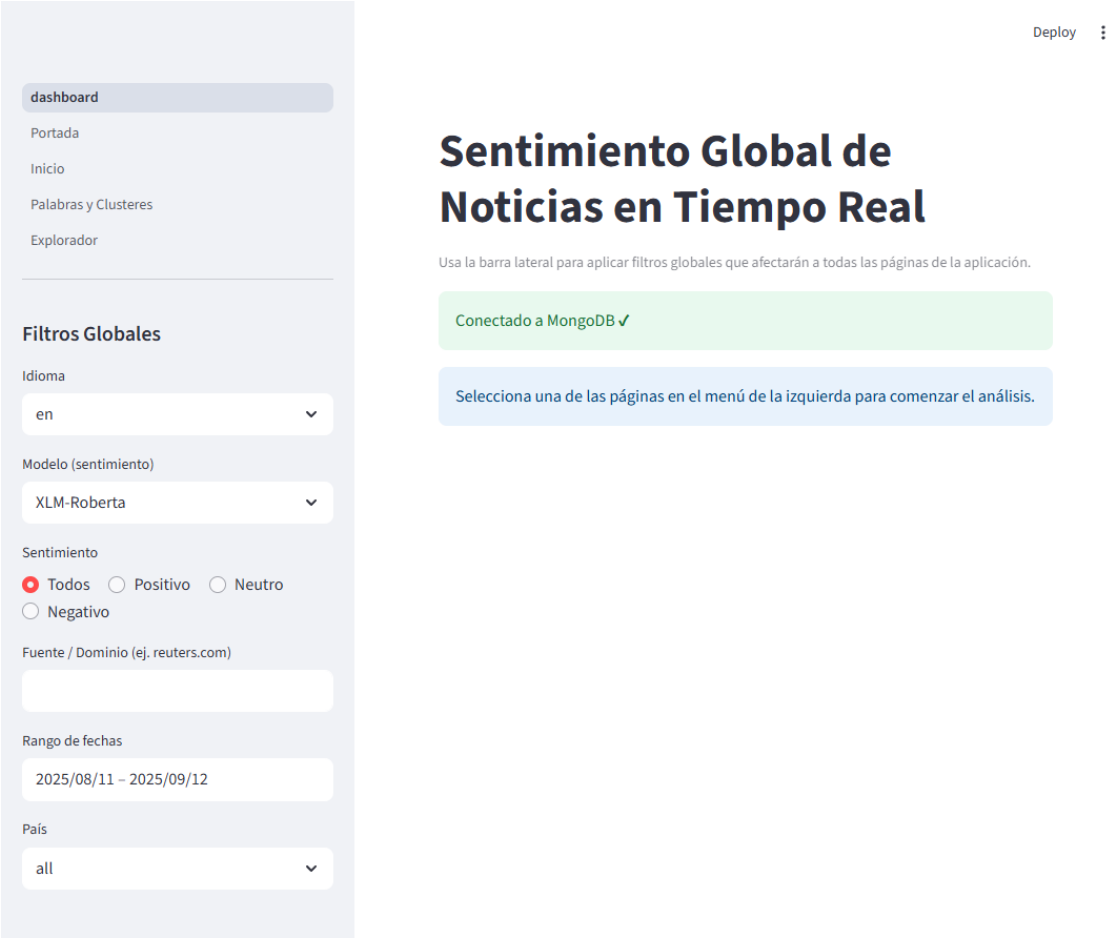
[23] P. Sánchez y M. López, “The Validity of Sentiment Analysis: Comparing Manual Annotation” Conf. Empir. Methods Nat. Lang. Process., 2023.

[24] K. Lee et al., “Sentiment Analysis Validation through Human Annotation” J. Mach. Learn. Res., vol. 21, 2023.

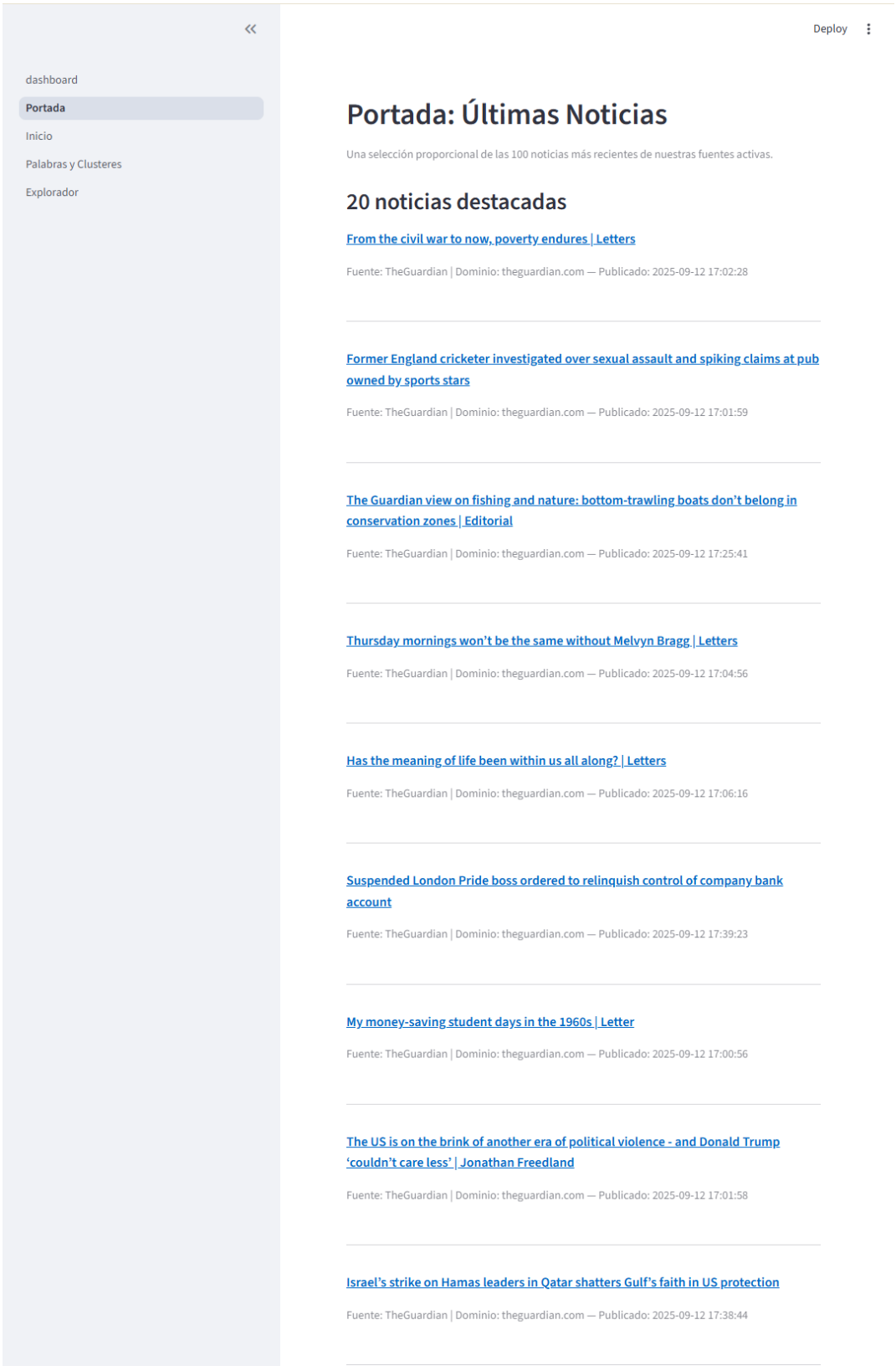
[25] J. Wang y L. Chen, “Correlating Automated Sentiment Scores with Self-Reported Attitude Scores” Psychol. Comput., 2024.

[26] D. Smith y A. Jones, “Human vs. Automated Sentiment Analysis: A Meta-Analysis” Front. Psychol., vol. 15, 2024.

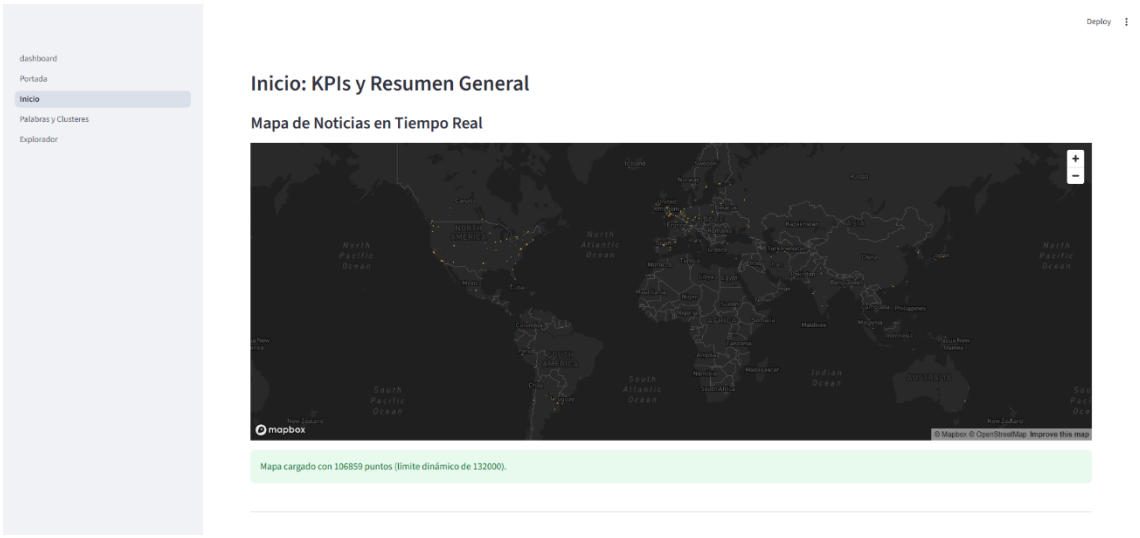
Capítulo 9. ANEXOS



Anexo 1. Streamlit – Dashboard

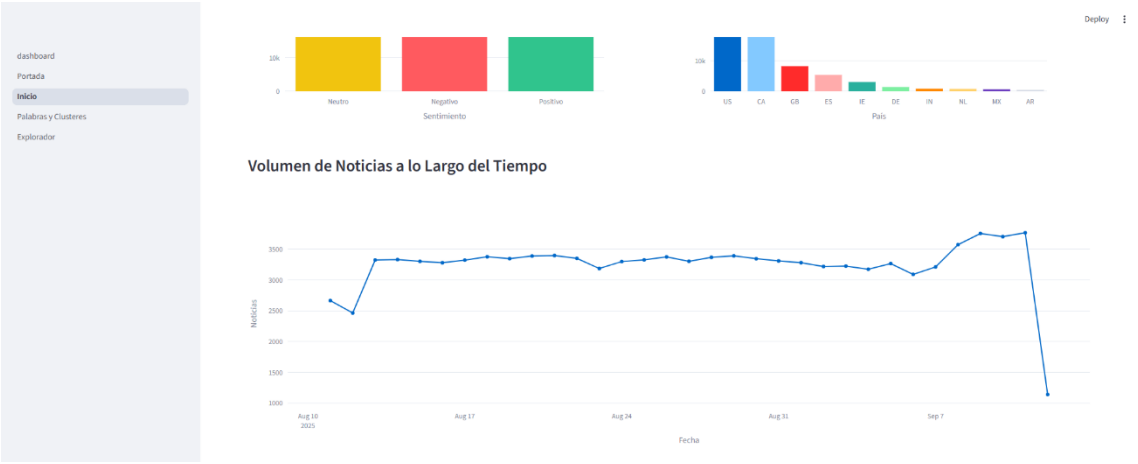


Anexo 2. Streamlit – Portada

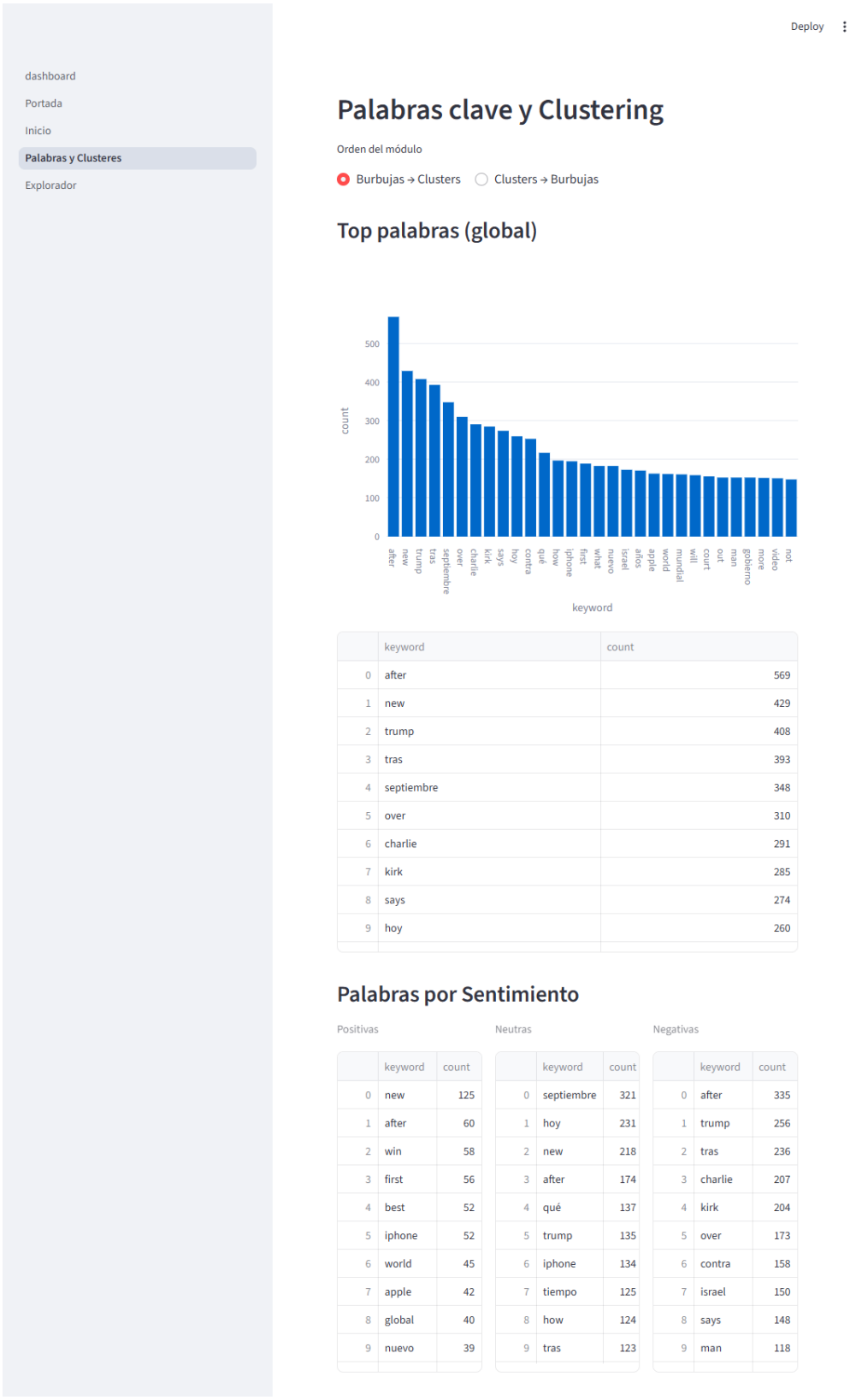


Anexo 3. Streamlit – Inicio: Mapa de Noticias en Tiempo Real

Anexo 4. Streamlit – Inicio: Indicadores Clave y Distribución de Sentimiento



Anexo 5. Streamlit – Inicio: Volumen de Noticias a lo Largo del Tiempo



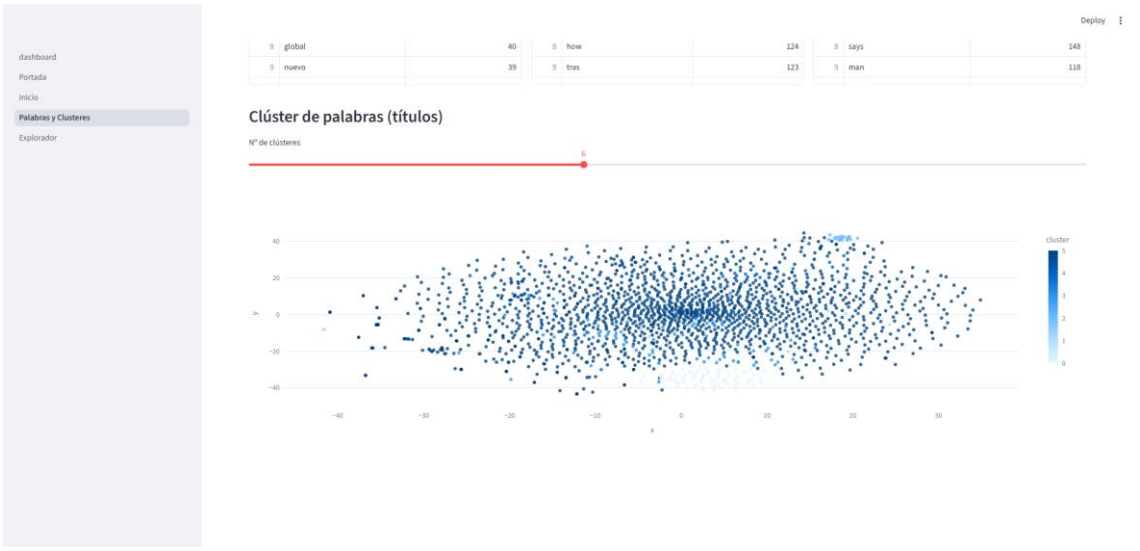
	keyword	count
0	after	569
1	new	429
2	trump	408
3	tras	393
4	septiembre	348
5	over	310
6	charlie	291
7	kirk	285
8	says	274
9	hoy	260

	keyword	count
0	new	125
1	after	60
2	win	58
3	first	56
4	best	52
5	iphone	52
6	world	45
7	apple	42
8	global	40
9	nuevo	39

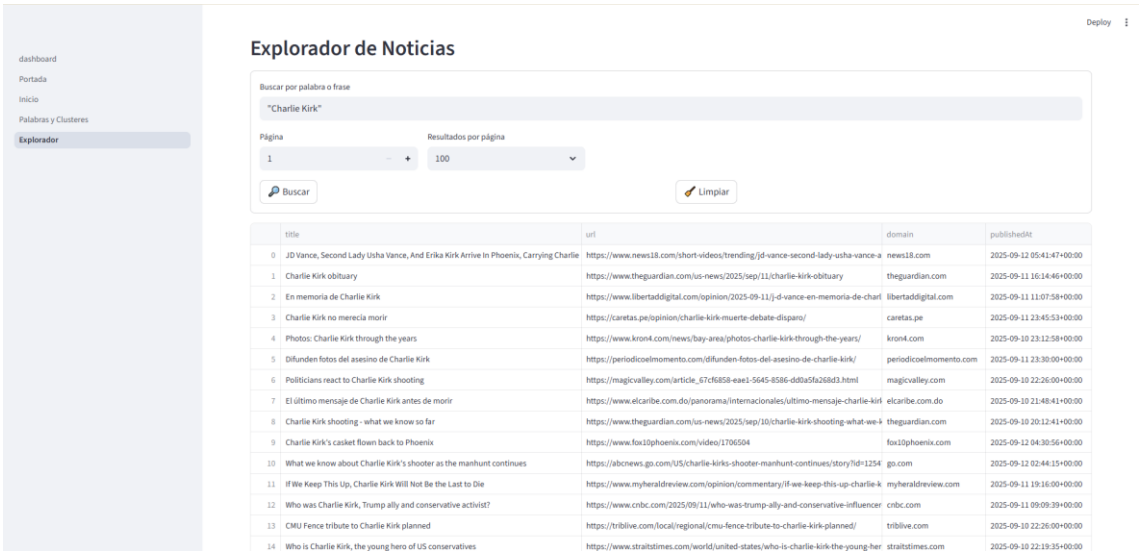
	keyword	count
0	septiembre	321
1	hoy	231
2	new	218
3	after	174
4	qué	137
5	trump	135
6	iphone	134
7	tiempo	125
8	how	124
9	tras	123

	keyword	count
0	after	335
1	trump	256
2	tras	236
3	charlie	207
4	kirk	204
5	over	173
6	contra	158
7	israel	150
8	says	148
9	man	118

Anexo 6. Streamlit – Palabras y Clústeres: Palabras



Anexo 7. Streamlit – Palabras y Clústeres: Clústeres



Anexo 8. Streamlit – Palabras y Clústeres: Clústeres

```yaml
services:
  mongodb01:
    image: mongo:5.0
    container_name: mongodb01
    ports:
      - "27073:27018"
    volumes:
      - ./data/mongodb01:/data/db
      - ./mongod01.conf:/etc/mongod.conf
      - ./mongo-keyfile:/run/secrets/mongo-keyfile
    command: >
      bash -c "
        cp /run/secrets/mongo-keyfile /etc/mongo-keyfile &&
        chmod 600 /etc/mongo-keyfile &&
        mongod --config /etc/mongod.conf
      "

  mongodb03:
    image: mongo:5.0
    container_name: mongodb03
    ports:
      - "27075:27020"
    volumes:
      - ./data/mongodb03:/data/db
      - ./mongod03.conf:/etc/mongod.conf
      - ./mongo-keyfile:/run/secrets/mongo-keyfile
    command: >
      bash -c "
        cp /run/secrets/mongo-keyfile /etc/mongo-keyfile &&
        chmod 600 /etc/mongo-keyfile &&
        mongod --config /etc/mongod.conf
      "

  backup:
    image: mongo:7
    container_name: mongo-backup
    volumes:
      - ./scripts:/scripts:ro
      - ./backups:/backup
    depends_on:
      - mongodb01
      - mongodb03
    env_file:
      - .env.backup
    entrypoint: ["sh", "-c", "while true; do /scripts/backup.sh; sleep 3600; done"]

  ingest-news-en-service:
    image: python:3.10-slim
    container_name: ingest-news-en-service
    volumes:
      - ./ingest/en:/app
      - ./logs:/logs
    working_dir: /app
    env_file:
      - .env.ingest_en
    command: >
      sh -c "pip install motor requests tldextract httpx && python ingest_schedule.py"
    depends_on:
      - mongodb01

  ingest-news-es-service:
    image: python:3.10-slim
    container_name: ingest-news-es-service
    volumes:
      - ./ingest/es:/app
      - ./logs:/logs
    working_dir: /app
    env_file:
      - .env.ingest_es
    command: >
      sh -c "pip install motor requests tldextract httpx && python ingest_schedule_es.py"
    depends_on:
      - mongodb01

  datascience-extract-service:
    image: python:3.10-slim
    container_name: datascience-extract-service
    volumes:
      - ./datascience:/app
      - ./logs/datascience:/logs/datascience
    working_dir: /app
    env_file:
      - .env.extract
    command: >
      sh -c "pip install -r requirements.txt && python extract_schedule.py"
    depends_on:
      - mongodb01
      - mongodb03

  title-cleaner-service:
    image: python:3.10-slim
    container_name: title-cleaner-service
    volumes:
      - ./models/title_cleaner:/app
      - ./logs/eval:/logs/eval
      - ./.env.eval_title_cleaner:/app/.env
    working_dir: /app
    command: >
      sh -c "pip install pymongo regex demoji python-dotenv && python title_cleaner.py"
    depends_on:
      - mongodb03
    restart: "no"

  dashboard:
    build:
      context: ./dashboard
      dockerfile: Dockerfile
    container_name: dashboard
    env_file:
```

Anexo 9. Extracto de docker-compose.yml

```python
import os
import requests
import motor.motor_asyncio
from datetime import datetime
import logging

api_name = "newsdata"
log_dir = os.getenv("LOG_FOLDER", "/logs")
os.makedirs(log_dir, exist_ok=True)
log_file = os.path.join(log_dir, f"{api_name}.log")

logger = logging.getLogger(api_name)
logger.setLevel(logging.INFO)

if not logger.handlers:
    handler = logging.FileHandler(log_file)
    formatter = logging.Formatter('%(asctime)s - %(levelname)s - %(message)s')
    handler.setFormatter(formatter)
    logger.addHandler(handler)
    logger.propagate = False

# NEWSDATA_API_KEY se obtiene exclusivamente del entorno
NEWSDATA_API_KEY = os.getenv("NEWSDATA_API_KEY")
if not NEWSDATA_API_KEY:
    logger.error("La variable de entorno NEWSDATA_API_KEY no está definida. Abortando ingesta de NewsData.io.")
    raise RuntimeError("NEWSDATA_API_KEY no definido")

# MONGO_URI se obtiene exclusivamente del entorno
MONGO_URI = os.getenv("MONGO_URI")
if not MONGO_URI:
    logger.error("La variable de entorno MONGO_URI no está definida. Abortando ingesta de NewsData.io.")
    raise RuntimeError("MONGO_URI no definido")

NEWSDATA_URL = "https://newsdata.io/api/1/news"

client = motor.motor_asyncio.AsyncIOMotorClient(MONGO_URI)
db = client["ingest-news"]
news_collection = db["news"]

async def ingest_newsdata():
    logger.info(f"[START] Ingesta iniciada desde {api_name}")
    inserted, skipped = 0, 0

    params = {
        "apikey": NEWSDATA_API_KEY,
        "language": "en"
    }

    try:
        response = requests.get(NEWSDATA_URL, params=params)
        if not response.ok:
            logger.error(f"[ERROR] Falló NewsData.io: {response.status_code} - {response.text}")
            return

        data = response.json()
        articles = data.get("results", [])
        if not articles:
            logger.info("[END] Sin artículos recibidos.")
            return

        logger.info(f"Artículos obtenidos: {len(articles)}")

        for art in articles:
            link = art.get("link")
            if not link:
                continue

            existing = await news_collection.find_one({
                "$or": [
                    {"link": link},
                    {"url": link},
                    {"webUrl": link}
                ]
            })
            if existing:
                skipped += 1
                continue

            document = {
                "article_id": art.get("article_id"),
                "title": art.get("title"),
                "link": art.get("link"),
                "keywords": art.get("keywords"),
                "creator": art.get("creator"),
                "description": art.get("description"),
                "pubDate": art.get("pubDate"),
                "pubDateTZ": art.get("pubDateTZ"),
                "image_url": art.get("image_url"),
                "source_id": art.get("source_id"),
                "source_name": art.get("source_name"),
                "source_priority": art.get("source_priority"),
                "source_url": art.get("source_url"),
                "source_icon": art.get("source_icon"),
                "language": art.get("language"),
                "country": art.get("country"),
                "category": art.get("category"),
                "datetime": datetime.utcnow().strftime("%Y%m%d %H%M%S"),
                "apiSource": "NewsData"
            }

            await news_collection.insert_one(document)
            inserted += 1
            logger.info(f"[INSERT] {link}")

        logger.info(f"[END] NewsData.io completado. Insertados: {inserted}, Repetidos: {skipped}")

    except Exception as e:
        logger.error(f"[ERROR] Error inesperado: {e}")
```

Anexo 10. Extracto de ingest_newsdata.py

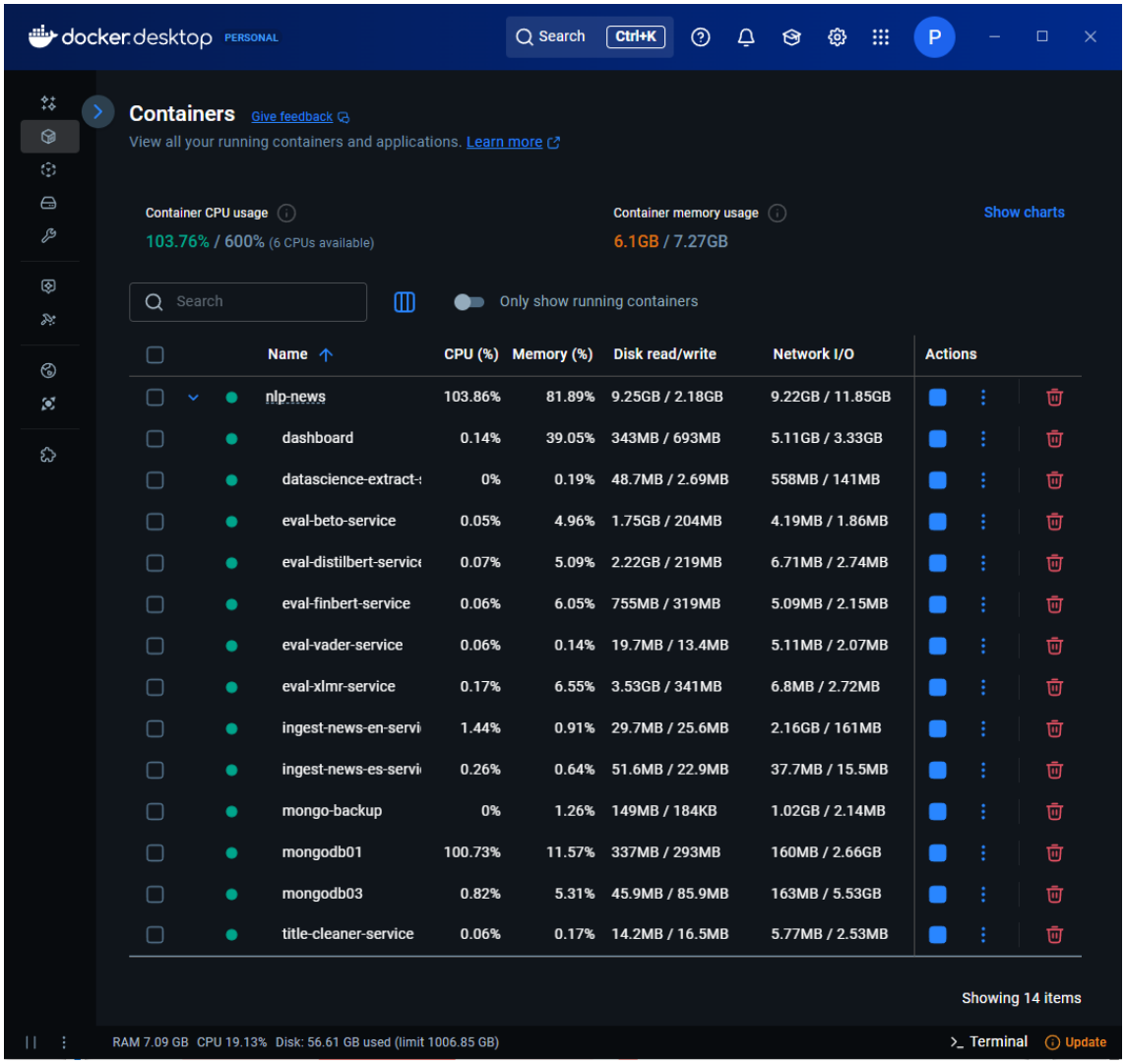
```
---

## 📁 Estructura del proyecto

A continuación, la estructura de carpetas y archivos en la raíz del repositorio:

```
.
├── .git/                              ← Control de versiones Git
├── .gitignore                         ← Ficheros/carpetas a ignorar en Git
├── ApiKeysES.txt                      ← (Ejemplo) Claves para APIs en español
├── credentials.txt                    ← (Ejemplo) Credenciales generales
├── docker-compose.yml                 ← Orquestación Docker Compose de todos los servicios
├── mongod01.conf                      ← Configuración de MongoDB (nodo primario)
├── mongod03.conf                      ← Configuración de MongoDB (nodo secundario)
├── mongo-keyfile                      ← Archivo de clave para Replica Set de MongoDB
├── README.md                          ← Este fichero (resumen y guía del proyecto)
│
├── backups/                           ← Copias de seguridad automáticas o manuales de MongoDB
│   ├── mongod01/                      ← Backups de cada nodo (p. ej. "2025-06-03_01", "2025-06-04_15", etc.)
│   └── mongod03/                      ← Backups de réplica secundaria con la misma estructura de fechas
│
├── dashboard/                         ← Código del Dashboard Streamlit
│   ├── app.py                         ← Punto de entrada de la aplicación
│   ├── requirements.txt               ← Dependencias de Python (Streamlit, pandas, requests, etc.)
│   └── …
│
├── data/                              ← Datos estáticos o dumps de ejemplo
│   ├── mongod01/                      ← Volumen de datos de MongoDB para el nodo 1
│   └── mongod03/                      ← Volumen de datos de MongoDB para el nodo 3
│
├── datascience/                       ← Notebooks y scripts de análisis/experimentos
│   ├── notebooks/
│   │   ├── exploracion_sentimiento.ipynb
│   │   └── entrenamiento_modelos.ipynb
│   └── utils/
│       └── helper_functions.py
│
├── ingest/                            ← Microservicios de ingesta de noticias
│   ├── gnews/                         ← Ingesta de noticias en español (GNews)
│   │   ├── Dockerfile
│   │   ├── ingest_gnews.py
│   │   ├── requirements.txt
│   │   └── .env.ingest_es             ← Variables de entorno para GNews (español)
│   │
│   ├── newsapi/                       ← Ingesta de noticias en inglés (NewsAPI)
│   │   ├── Dockerfile
│   │   ├── ingest_newsapi.py
│   │   ├── requirements.txt
│   │   └── .env.ingest_en             ← Variables de entorno para NewsAPI (inglés)
│   │
│   ├── newsdata/                      ← Ingesta desde NewsData.io
│   │   ├── Dockerfile
│   │   ├── ingest_newsdata.py
│   │   ├── requirements.txt
│   │   └── .env.extract               ← Variables de entorno para NewsData.io
```

Anexo 11. Extracto de README.md
```



Anexo 12. Solución nlp-news

```
ingest-news> db.news.find({
...   "evaluatedMedia.checkedAt": {
...     $gte: ISODate("2025-07-01T00:00:00Z"),
...     $lt:  ISODate("2025-07-08T00:00:00Z")
...   }
... }).hint({"evaluatedMedia.checkedAt":1}).explain("executionStats")
{

ingest-news> db.news.find({
...   "evaluatedMedia.checkedAt": {
...     $gte: ISODate("2025-07-01T00:00:00Z"),
...     $lt:  ISODate("2025-07-08T00:00:00Z")
...   }
... }).hint({"evaluatedMedia.checkedAt":1}).explain("executionStats")
{
  explainVersion: '1',
  queryPlanner: {
    namespace: 'ingest-news.news',
    indexFilterSet: false,
    parsedQuery: {
      '$and': [
        {
          'evaluatedMedia.checkedAt': { '$lt': ISODate('2025-07-08T00:00:00.000Z') }
        },
        {
          'evaluatedMedia.checkedAt': { '$gte': ISODate('2025-07-01T00:00:00.000Z') }
        }
      ]
    },
    maxIndexedOrSolutionsReached: false,
    maxIndexedAndSolutionsReached: false,
    maxScansToExplodeReached: false,
    winningPlan: {
      stage: 'FETCH',
      inputStage: {
        stage: 'IXSCAN',
        keyPattern: { 'evaluatedMedia.checkedAt': 1 },
        indexName: 'evaluatedMedia.checkedAt_1',
        isMultiKey: false,
        multiKeyPaths: { 'evaluatedMedia.checkedAt': [] },
        isUnique: false,
        isSparse: false,
        isPartial: false,
        indexVersion: 2,
        direction: 'forward',
        indexBounds: {
          'evaluatedMedia.checkedAt': [ '[new Date(1751328000000), new Date(1751932800000))' ]
        }
      }
    },
    rejectedPlans: []
  },
  executionStats: {
    executionSuccess: true,
    nReturned: 0,
    executionTimeMillis: 0,
    totalKeysExamined: 0,
    totalDocsExamined: 0,
    executionStages: {
      stage: 'FETCH',
      nReturned: 0,
      executionTimeMillisEstimate: 0,
      works: 1,
      advanced: 0,
      needTime: 0,
      needYield: 0,
      saveState: 0,
      restoreState: 0,
      isEOF: 1,
      docsExamined: 0,
      alreadyHasObj: 0,
      inputStage: {
        stage: 'IXSCAN',
        nReturned: 0,
        executionTimeMillisEstimate: 0,
        works: 1,
        advanced: 0,
        needTime: 0,
```

Anexo 13. Consulta utilizando índice

```
ingest-news> db.news.find({
...   "evaluatedMedia.checkedAt": {
...     $gte: ISODate("2025-07-01T00:00:00Z"),
...     $lt:  ISODate("2025-07-08T00:00:00Z")
...   }
... }).hint({$natural:1}).explain("executionStats")
{
  explainVersion: '1',
  queryPlanner: {
    namespace: 'ingest-news.news',
    indexFilterSet: false,
    parsedQuery: {
      '$and': [
        {
          'evaluatedMedia.checkedAt': { '$lt': ISODate('2025-07-08T00:00:00.000Z') }
        },
        {
          'evaluatedMedia.checkedAt': { '$gte': ISODate('2025-07-01T00:00:00.000Z') }
        }
      ]
    },
    maxIndexedOrSolutionsReached: false,
    maxIndexedAndSolutionsReached: false,
    maxScansToExplodeReached: false,
    winningPlan: {
      stage: 'COLLSCAN',
      filter: {
        '$and': [
          {
            'evaluatedMedia.checkedAt': { '$lt': ISODate('2025-07-08T00:00:00.000Z') }
          },
          {
            'evaluatedMedia.checkedAt': { '$gte': ISODate('2025-07-01T00:00:00.000Z') }
          }
        ]
      },
      direction: 'forward'
    },
    rejectedPlans: []
  },
  executionStats: {
    executionSuccess: true,
    nReturned: 0,
    executionTimeMillis: 208,
    totalKeysExamined: 0,
    totalDocsExamined: 254370,
    executionStages: {
      stage: 'COLLSCAN',
      filter: {
        '$and': [
          {
            'evaluatedMedia.checkedAt': { '$lt': ISODate('2025-07-08T00:00:00.000Z') }
          },
          {
            'evaluatedMedia.checkedAt': { '$gte': ISODate('2025-07-01T00:00:00.000Z') }
          }
        ]
      },
      nReturned: 0,
      executionTimeMillisEstimate: 69,
      works: 254372,
      advanced: 0,
      needTime: 254371,
      needYield: 0,
      saveState: 254,
      restoreState: 254,
      isEOF: 1,
      direction: 'forward',
      docsExamined: 254370
    }
```

Anexo 14. Consulta sin utilizar índice