



MÁSTER UNIVERSITARIO DE BIG DATA

TRABAJO FIN DE MÁSTER

Sistema Inteligente de Análisis de Sentimientos de Mercados Financieros

Autor: Blanca María de Pedro Morejón

Director: Miguel Ángel Sanz Bobi

Madrid

Junio de 2025

Declaro, bajo mi responsabilidad, que el Proyecto presentado con el título
Sistema Inteligente de Análisis de Sentimientos de Mercados
Financieros

en la ETS de Ingeniería - ICAI de la Universidad Pontificia Comillas en el
curso académico 2024/2025. es de mi autoría, original e inédito y
no ha sido presentado con anterioridad a otros efectos. El Proyecto no es
plagio de otro, ni total ni parcialmente y la información que ha sido tomada
de otros documentos está debidamente referenciada.

Blanca de Pedro Morejón

Fdo.: Blanca María de Pedro Morejón

Fecha: 02/ 06/ 2025

Autorizada la entrega del proyecto

EL DIRECTOR DEL PROYECTO

Fdo.: Miguel Ángel Sanz Bobi

Fecha: 02/ 06/ 2025

RESUMEN Y PALABRAS CLAVE

RESUMEN

Este trabajo presenta el desarrollo de una plataforma interactiva para el análisis del sentimiento expresado en titulares financieros y la visualización de su impacto sobre el comportamiento del mercado. Se han aplicado diferentes enfoques de procesamiento del lenguaje natural (NLP), desde modelos clásicos basados en diccionarios hasta redes neuronales LSTM y modelos transformers especializados como FinBERT y FinancialBERT. Además, se ha realizado el *fine-tuning* de varios modelos para adaptarlos mejor al contexto financiero, integrándolos después en un *ensemble* que combina sus predicciones.

Una parte clave del proyecto ha sido evaluar cuál de estos modelos ofrece mejores resultados y cómo de bien son capaces de clasificar titulares como positivos, negativos o neutros. A partir de ahí, se ha desarrollado una interfaz interactiva en Streamlit que permite al usuario explorar la evolución del sentimiento por empresa y periodo temporal, realizar análisis de tópicos por clase emocional y generar resúmenes automáticos a partir de los titulares. También se ha cruzado este análisis textual con datos reales del mercado (precios y volumen de acciones), generando una capa adicional de *insights* para entender mejor qué está pasando y por qué.

Los resultados muestran una mejora significativa en la clasificación de sentimientos al aplicar *fine-tuning* y estrategias de combinación de modelos, lo que respalda la viabilidad del sistema para tareas reales de análisis financiero.

PALABRAS CLAVE

Análisis de sentimientos, noticias financieras, modelos BERT, NLP, transformers, fine-tuning, aprendizaje profundo, visualización.

ABSTRACT AND KEYWORDS

ABSTRACT

This project presents the development of an interactive platform for analyzing the sentiment expressed in financial news headlines and visualizing its impact on market behavior. Different natural language processing (NLP) approaches have been applied, ranging from traditional dictionary-based models to LSTM neural networks and specialized transformer models such as FinBERT and FinancialBERT. In addition, several models have been fine-tuned to better adapt to the financial context and then integrated into an ensemble that combines their predictions.

A key part of the project was evaluating which of these models performs best and how accurately they can classify headlines as positive, negative, or neutral. Based on this, an interactive Streamlit interface was developed, allowing users to explore sentiment trends by company and time period, perform topic modeling by sentiment class, and generate automatic summaries from the headlines. This textual analysis was also cross-referenced with real market data (stock prices and volumes), adding an extra layer of insights to better understand what is happening and why.

The results show a significant improvement in sentiment classification when applying fine-tuning and ensemble strategies, supporting the system's viability for real-world financial analysis tasks.

KEYWORDS

Sentiment analysis, financial news, BERT models, NLP, transformers, fine-tuning, deep learning, visualization.

CONTENIDOS

Resumen y Palabras Clave	1
Resumen.....	1
Palabras Clave.....	1
Abstract and Keywords	2
Abstract.....	2
Keywords.....	2
Índice de Tablas	6
Índice de Ilustraciones	7
Capítulo 1. Estado de la Cuestión	9
Contexto del Sector Financiero.....	9
Lenguajes de Procesamiento Natural.....	11
Implementación de NLP en el Sector Financiero	12
Capítulo 2. Definición del Proyecto	15
Motivación.....	16
Objetivos.....	17
Metodología.....	18
Herramientas Utilizadas.....	19
Estructura del Trabajo y Planificación.....	20
Capítulo 3. Obtención de los Datos	21
Noticias.....	21
Dataset con etiquetas de sentimiento	21
Dataset principal: titulares con información de mercado	22
Stock	23
Pre-procesamiento de datos	25
Capítulo 4. Análisis de Sentimientos	27
Enfoque basado en Diccionarios.....	28

<i>Loughran-McDonald</i>	28
<i>VADER</i>	32
<i>Comparación de las Métricas de Rendimiento</i>	33
Modelos LSTM.....	36
<i>LSTM Básico</i>	36
<i>LSTM junto con GloVe</i>	38
<i>LSTM junto con FastText</i>	39
<i>Comparación de las Métricas de Rendimiento</i>	41
Bert 43	
<i>FinBert Clásico</i>	44
<i>FinBert Junto a PipeLine de HuggingFace</i>	46
<i>Fin-tuning de FinBert</i>	48
<i>FinancialBERT</i>	49
<i>Fin-tuning de FinancialBERT</i>	51
LLM53	
Ensemble.....	54
Capítulo 5. Generación de Informes	57
Sistema de Filtrado	59
KeyWords	60
Sentiment Analysis	61
Topic Modeling.....	63
Evolución Temporal del Mercado.....	65
Resumen de las Ideas Principales	67
Integración Final en la Interfaz	68
Capítulo 6. Análisis de Resultados	70
Evaluación de Modelos de Clasificación de Sentimiento	70
<i>Modelos clásicos</i>	70
<i>Modelos LSTM</i>	71
<i>Modelos basados en transformers</i>	71
<i>Ensemble</i>	72
<i>Comparación visual</i>	73

Análisis de Resultados de la Generación de Informes	74
<i>Diseño de la interfaz</i>	74
<i>Funcionalidades principales y Tecnología</i>	Error! Bookmark not defined.
<i>Funcionamiento y características clave</i>	78
<i>Valoración global del sistema</i>	85
Capítulo 7. Conclusiones Finales	87
Trabajos Futuros	88
<i>Recuperación de datos de mercado para tickers problemáticos</i>	88
<i>Aplicación de modelos LLM para análisis de sentimiento</i>	89
<i>Simulación de la evolución del mercado basada en sentimiento</i>	89
Capítulo 8. Bibliografía.....	90
ANEXOS 97	
Anexo A: ODS.....	97
<i>ODS 8 – Trabajo decente y crecimiento económico</i>	97
<i>ODS 9 – Industria, innovación e infraestructura</i>	97
<i>ODS 17 – Alianzas para lograr los objetivos</i>	98
Anexo B: Extracción de Datos Financieros	98
Anexo C: Métricas de Evaluación	101
<i>Accuracy</i>	101
<i>Precision (Precisión)</i>	101
<i>Recall (Sensibilidad)</i>	102
<i>F1 Score</i>	102
<i>Matriz de Confusión</i>	102
Anexo D: Código y estructura de Streamlit.....	103
<i>Análisis de Titulares</i>	104
<i>Análisis de Sentimientos</i>	105
<i>Topic Modeling</i>	106
<i>Tendencias de Mercado</i>	107

ÍNDICE DE TABLAS

Table 1: Planificación temporal del Trabajo Fin de Máster	20
Table 2: Dataset con etiquetas de sentimiento	21
Table 3: Estructura del dataset de titulares financieros	22
Table 4: Estructura del dataset histórico de precios bursátiles.....	24
Table 5: Interpretación del sentiment score básico	29
Table 6: Comparación global del rendimiento entre los modelos basados en diccionario..	34
Table 7: Comparativa del rendimiento por clase negative	35
Table 8: Comparativa del rendimiento por clase neutral.....	35
Table 9: Comparativa del rendimiento por clase neutral.....	35
Table 10: Métricas de Rendimiento LSTM Bidireccional	37
Table 11: Métricas de Rendimiento LSTM + GloVe.....	38
Table 12: Métricas de Rendimiento LSTM + FastTest.....	40
Table 13: Comparación global del rendimiento entre los modelos basados en LSTM.....	41
Table 14: Comparativa del rendimiento por clase negative	42
Table 15: Comparativa del rendimiento por clase neutral.....	42
Table 16: Comparativa del rendimiento por clase positive	43
Table 17: Resultados de clasificación de FinBERT manual	46
Table 18: Resultados de clasificación de FinBERT con Pipeline.....	47
Table 19: Resultados de clasificación de FinBERT con fine-tuning	49
Table 20: Resultados de clasificación de FinancialBERT	50
Table 21: Resultados de clasificación de FinancialBERT con Fin-Tuning	51
Table 22: Resultados de clasificación de Ensemble.....	55

ÍNDICE DE ILUSTRACIONES

Ilustración 1: Aplicación del diccionario Loughran-McDonald.....	30
Ilustración 2: Comparativa de resultados para diferentes variantes de sentiment score con umbral ± 0.1	31
Ilustración 3: Evaluación del score básico vs. ajustado mediante métricas de clasificación por clase.....	32
Ilustración 4: Comparación de los resultados obtenidos por las técnicas basadas en diccionarios.....	34
Ilustración 5: Resultados del modelo FinBERT	45
Ilustración 6: Matriz de confusión del modelo ensemble basado en votación mayoritaria. 56	
Ilustración 7: Diagrama de componentes de la aplicación	58
Ilustración 8: Ejemplo sistema de filtrado	59
Ilustración 9: Interfaz de Análisis de Titulares.....	60
Ilustración 10: Interfaz Análisis de Sentimiento	61
Ilustración 11: Evolución temporal del número de titulares clasificados por sentimiento..	62
Ilustración 12: Evolución del sentimiento medio diario a partir del análisis de titulares financieros.	62
Ilustración 13: Top 5 días con sentimiento medio más positivo y más negativo según los titulares financieros.....	63
Ilustración 14: Interfaz Topic Modeling por Sentimiento	65
Ilustración 15: Reresentación visual de las Tecendencia de Mercado	66
Ilustración 16: Comparación temporal Precio vs Sentimiento	66
Ilustración 17: Resumen automático generado a partir de los titulares financieros	67
Ilustración 18: Interfaz de Inicio	69
Ilustración 19: Matriz de confusión del enfoque basado en diccionarios.....	71
Ilustración 20: Matriz de confusión del enfoque basado en LSTM	71
Ilustración 21: Matriz de confusión del enfoque basado en BERT	72

Ilustración 22: Matriz de confusión del enfoque basado en BERT con Fin-Tuning.....	72
Ilustración 23: Matriz de confusión del enfoque basado en Ensemble	72
Ilustración 24: Accuracy por modelo de clasificación de sentimiento.....	73
Ilustración 25: F1 ponderado por modelo de clasificación de sentimiento.	73
Ilustración 26: Ejemplo del Diseño de la Interfaz	74
Ilustración 27: Ejemplo de mensaje visual mostrado durante la carga del módulo	75
Ilustración 28: Ejemplo de interacción con un gráfico.....	76
Ilustración 29: Ejemplo de las opción adicionales que ofrecen los gráficos.....	76
Ilustración 30: Ejemplo del panel lateral de la interfaz.....	77
Ilustración 31: Ejemplo de un mensaje informativo por pantalla.....	77
Ilustración 32: Análisis de Titulares de Tesla (2015-2020)	79
Ilustración 33: Distribución general del sentimiento de Tesla (2015-2020)	80
Ilustración 34: Evolución temporal del sentimiento de Tesla (2015-2020)	80
Ilustración 35: Sentimiento medio diario de Tesla (2015-2020).....	81
Ilustración 36: Interfaz de inicio de Topic Modeling.....	81
Ilustración 37: Generación de tópicos con sentimiento positivo.....	82
Ilustración 38: Generación de tópicos con sentimiento neutral.....	82
Ilustración 39: Generación de tópicos con sentimiento negativo	83
Ilustración 40: Interfaz de Tendencias de Mercado de Tesla (2015-2020).....	84
Ilustración 41: Cierre de la Interfaz de Tendencias de Mercado.....	85
Ilustración 42: Ejemplo matriz de confusión de 3 clases	103

Capítulo 1. ESTADO DE LA CUESTIÓN

CONTEXTO DEL SECTOR FINANCIERO

El sector financiero ha experimentado en las últimas décadas una rápida transformación digital que ha modificado la forma en que operan analistas, inversores e instituciones. Se ha reemplazado progresivamente la operativa tradicional por plataformas digitales automatizadas (banca online, aplicaciones móviles, y trading electrónico). Según PwC (2020), el 89% de las entidades financieras han priorizado la transformación digital como estrategia clave [1].

El aumento en la velocidad y el volumen de la información obliga a gestionar grandes cantidades de datos en tiempo real. Cada día se generan cantidades masivas de información procedentes de canales como redes sociales, medios de comunicación y plataformas financieras. Por ejemplo, en Twitter se publican alrededor de 500 millones de tuits diarios a nivel mundial [2]. Al mismo tiempo, han aparecido nuevos actores y tecnologías que están cambiando el sector financiero [3]. El uso de estrategias automatizadas, como el trading de alta frecuencia (HFT), se ha vuelto cada vez más común en los mercados bursátiles. Estas técnicas emplean algoritmos complejos y sistemas avanzados para ejecutar operaciones a velocidades extremadamente altas, permitiendo a las instituciones financieras capitalizar pequeñas fluctuaciones de precios en fracciones de segundo. El HFT, en particular, ha ganado popularidad entre inversores, utilizando algoritmos matemáticos sofisticados para realizar grandes volúmenes de transacciones de forma automática [4].

Estas tendencias reflejan cómo las innovaciones tecnológicas (inteligencia artificial, HPC, etc.) están redefiniendo el sector financiero, proporcionando mayor velocidad, eficiencia y capacidades analíticas avanzadas en las decisiones que toman. Si bien los datos numéricos (precios, indicadores, transacciones) siguen siendo esenciales en finanzas, gran parte del valor informativo reside en datos no estructurados, como noticias, informes corporativos,

publicaciones en redes sociales, comunicados de resultados, etc. Se calcula que más del 80% de los datos generados en servicios financieros son de carácter no estructurado [5] (documentos, comunicaciones o comentarios), y contienen información crítica que puede ser desaprovechada debido a su complejidad de análisis.

El contenido textual permite revelar tendencias, realizar análisis de sentimiento del mercado y descubrir patrones que los enfoques cuantitativos tradicionales podrían pasar por alto. Las empresas que integran datos textuales no estructurados en sus procesos de decisión han llegado a mejorar hasta un 30% su retorno de inversión, y son un 23% más propensas a superar a sus competidores en crecimiento e innovación [6].

Existe un consenso en que las herramientas capaces de interpretar automáticamente el lenguaje natural son fundamentales para complementar el análisis financiero. Técnicas de Procesamiento del Lenguaje Natural (NLP, por sus siglas en inglés) y análisis de sentimientos se han incorporado para analizar grandes volúmenes de texto, permitiendo a las instituciones financieras captar el sentimiento del mercado y tomar decisiones de inversión mejor informadas. De hecho, algunos de los fondos cuantitativos más avanzados (por ejemplo, Two Sigma) emplean ya algoritmos de NLP para analizar millones de mensajes en redes sociales e identificar opiniones positivas o negativas sobre empresas, incorporando estas señales en sus estrategias de trading [6].

LENGUAJES DE PROCESAMIENTO NATURAL

El procesamiento del lenguaje natural (Natural Language Processing o NLP) es un campo de la inteligencia artificial y la informática que busca que las máquinas sean capaces de interpretar y generar lenguaje humano de forma inteligente. El NLP pretende convertir información no estructurada, como texto, imágenes o audio, en estructurada e interpretable por sistemas informáticos. Se trata de un ámbito interdisciplinario que combina aspectos de la ciencia de datos, la lingüística computacional y el Machine Learning, con el objetivo de permitir la interacción entre los humanos y las máquinas a través del lenguaje [7].

A lo largo del tiempo, el NLP ha evolucionado desde sistemas basados en reglas (1950-1980), que requerían definir manualmente patrones, hasta modelos estadísticos (1990-2000), que usaban probabilidades y grandes volúmenes de texto para hacer predicciones. Técnicas como los n-gramas, TF-IDF o los clasificadores bayesianos permitieron avances en tareas como clasificación de texto o detección de spam. A partir de 2013, se produjo un salto importante con el uso de word embeddings (como Word2Vec y GloVe), que representaban palabras como vectores numéricos según su contexto, mejorando significativamente la comprensión semántica [8].

El cambio más relevante llegó con la aparición de los modelos Transformer en 2017, como BERT y GPT. Estos modelos permiten captar el significado completo de una frase considerando el contexto a ambos lados de cada palabra. Gracias a ellos, el NLP ha avanzado a un nuevo nivel. Ahora es posible analizar sentimientos, resumir textos, traducir entre idiomas o generar texto de forma coherente sin entrenar un modelo desde cero para cada tarea [8]. Modelos como GPT-3 o GPT-4 han demostrado capacidades excepcionales, incluso en tareas complejas como responder preguntas, redactar informes o interpretar lenguaje informal.

Actualmente, el NLP se aplica en muchas áreas con resultados destacables, como la clasificación de texto (por tema, tono, intención), el análisis de sentimiento, la detección de entidades como nombres de empresas o lugares (NER), el resumen automático o la generación de contenido. Su principal ventaja es que permite analizar grandes volúmenes de texto en poco tiempo, captando el contexto, las emociones y los temas principales sin intervención humana directa. Esto es especialmente útil en sectores donde se generan enormes cantidades de texto cada día, como los medios, las redes sociales o las finanzas [9].

Sin embargo, el NLP todavía enfrenta retos. La ambigüedad del lenguaje, la ironía o el sarcasmo son difíciles de interpretar para humanos, y más aún para sistemas automáticos. Además, los modelos actuales suelen estar entrenados mayoritariamente en inglés, dejando de lado a idiomas menos significativos. A su vez, un factor limitante es el coste computacional y el consumo energético de los grandes modelos, así como la transparencia y explicabilidad, ya que muchos funcionan como una “caja negra”, lo que dificulta saber por qué toman ciertas decisiones. Otro problema relevante es el sesgo presente en los datos con los que fueron entrenados, lo que puede afectar a la equidad de los resultados [10].

Aun con estos desafíos, el NLP se ha posicionado como una herramienta clave para transformar texto en conocimiento. Su uso en sectores como el financiero, donde el análisis de información no estructurada es crítico, representa una gran oportunidad para mejorar la toma de decisiones basada en datos. En la siguiente sección se explora cómo estas técnicas se están aplicando en finanzas, y cómo pueden ayudar a interpretar y anticipar el comportamiento del mercado a partir del análisis automático de noticias y publicaciones.

IMPLEMENTACIÓN DE NLP EN EL SECTOR FINANCIERO

El procesamiento del lenguaje natural (NLP) ha experimentado un crecimiento significativo en el sector financiero en los últimos años, facilitando el análisis automatizado de grandes volúmenes de texto. Su aplicación se ha integrado en diversos procesos financieros,

abarcando desde la automatización de tareas administrativas hasta el análisis avanzado de mercados. En este contexto, destacan las siguientes aplicaciones clave [\[11\]](#) [\[12\]](#):

1. **Análisis de sentimientos en mercados financieros:** La aplicación del análisis de sentimientos permite evaluar la percepción del mercado a partir de noticias, informes financieros y redes sociales. Por ejemplo, Two Sigma ha utilizado técnicas de NLP para analizar las actas del Comité Federal de Mercado Abierto (FOMC) y extraer información sobre las preocupaciones del mercado [\[13\]](#).
2. **Automatización de informes y generación de contenido:** NLP ha revolucionado la generación de informes financieros al automatizar la extracción y el resumen de información relevante, reduciendo tiempos de análisis y aumentando la precisión en la presentación de datos. Un caso concreto es Bloomberg, que ha desarrollado BloombergGPT, un modelo de lenguaje específico para finanzas, que mejora tareas como el análisis de sentimientos y la generación de informes [\[14\]](#).
3. **Clasificación automática de documentos financieros:** La capacidad de NLP para organizar y categorizar contratos, informes y otros documentos ha optimizado la gestión documental en el sector financiero, permitiendo una recuperación eficiente de la información.
4. **Predicción de tendencias de mercado a partir de datos textuales:** Una de las aplicaciones más prometedoras de NLP es su combinación con modelos de aprendizaje profundo para prever tendencias de mercado en función del análisis de noticias y métricas económicas. Estudios recientes han demostrado que el análisis de sentimientos de noticias financieras puede mejorar las predicciones de movimientos del mercado [\[40\]](#).

A pesar de los avances en NLP aplicado a las finanzas, existen diversos desafíos que se deben abordar para mejorar la precisión y fiabilidad de los modelos. Uno de los principales retos es la calidad y el sesgo de los datos, ya que la precisión de los modelos de NLP depende en gran medida de la calidad de los datos utilizados en su entrenamiento. La presencia de sesgo en los datos puede generar predicciones erróneas, lo que es particularmente

problemático en finanzas, donde la toma de decisiones basada en información defectuosa puede tener consecuencias económicas significativas. Con el objetivo de mitigar estos problemas, se están implementando técnicas avanzadas de limpieza y balanceo de datos, así como estrategias de preprocesamiento, como FinBERT y BloombergGPT, que han demostrado mejorar la precisión en tareas de clasificación y análisis de sentimiento [15].

Otro desafío clave radica en la privacidad y seguridad de la información. Los modelos de NLP procesan grandes volúmenes de texto, incluyendo noticias financieras, publicaciones en redes sociales y documentos regulatorios, lo que exige garantizar el cumplimiento normativo. Además, la interpretabilidad y explicabilidad de los modelos de NLP en finanzas sigue siendo un reto fundamental. Aunque los modelos basados en transformers, como BERT y GPT-3, han mejorado significativamente el análisis de textos financieros, su naturaleza de “caja negra” dificulta la comprensión de las decisiones que toman [15]. Esto es especialmente problemático en entornos financieros altamente regulados, donde se requiere justificar las decisiones tomadas por los modelos. Para abordar esta cuestión, se han desarrollado modelos híbridos que combinan NLP con reglas basadas en conocimiento financiero explícito [16].

Otro aspecto clave para mejorar la utilidad del NLP en finanzas es su integración con modelos de predicción cuantitativa. Aunque el análisis de sentimientos proporciona información valiosa sobre la percepción del mercado, su efectividad aumenta significativamente cuando se combina con modelos cuantitativos de predicción de precios y tendencias financieras [16]. En este sentido, los enfoques multimodales, que combinan NLP con datos estructurados como series temporales financieras, han mostrado ser prometedores al mejorar la capacidad predictiva de los modelos de inversión y gestión de riesgos [17]. Por ejemplo, el uso de transformers especializados para integrar texto con datos de mercado en tiempo real ha permitido mejorar la precisión en la detección de eventos financieros relevantes y su impacto en los precios de activos.

Capítulo 2. DEFINICIÓN DEL PROYECTO

En los últimos años, el sector financiero ha experimentado una transformación impulsada por el crecimiento exponencial de los datos y la necesidad de extraer información valiosa de fuentes no estructuradas, como noticias, informes corporativos y redes sociales. En este contexto, el Procesamiento del Lenguaje Natural (NLP, por sus siglas en inglés) ha emergido como una herramienta clave para analizar grandes volúmenes de información, facilitando el análisis financiero y mejorando la toma de decisiones estratégicas [\[18\]](#).

El uso de NLP en finanzas se ha expandido significativamente, permitiendo la automatización de tareas como la clasificación de documentos, el análisis de sentimiento y la generación de informes financieros. En particular, la capacidad de identificar el tono de las noticias financieras y el impacto que los acontecimientos descritos en ellas tienen en el mercado ha abierto nuevas oportunidades en la predicción de tendencias bursátiles [\[19\]](#). Modelos avanzados de NLP, como BERT y FinBERT, han sido optimizados específicamente para el análisis financiero, logrando interpretar de manera más precisa el lenguaje técnico de la industria [\[20\]](#) [\[21\]](#).

A pesar de estos avances, la integración de NLP con modelos de predicción de mercados sigue siendo un desafío. Si bien el análisis de sentimientos proporciona información relevante sobre la percepción del mercado, su combinación con métricas financieras cuantitativas es fundamental para mejorar la precisión de las predicciones. En este sentido, el desarrollo de sistemas que combinen NLP con algoritmos de aprendizaje profundo (Deep Learning) representa un campo de investigación necesario en la intersección de la inteligencia artificial y las finanzas [\[22\]](#) [\[23\]](#).

El presente proyecto tiene como objetivo desarrollar un sistema basado en NLP y aprendizaje profundo para analizar el impacto de las noticias en el mercado financiero y predecir

tendencias a partir de los datos obtenidos, con el propósito de facilitar la interpretación de la información y mejorar la toma de decisiones en el ámbito financiero.

MOTIVACIÓN

En el ámbito financiero, la información juega un papel crucial en la toma de decisiones, la capacidad de analizar grandes volúmenes de datos textuales en tiempo real se ha convertido en un factor clave para la toma de decisiones estratégicas. Noticias financieras, informes de empresas, redes sociales y publicaciones económicas tienen un impacto directo en los mercados, influyendo en la percepción de los inversores y, en última instancia, en la evolución de los precios de los stocks. Sin embargo, interpretar y extraer conocimiento representa un desafío significativo, especialmente dada la velocidad con la que la información puede cambiar y la complejidad del lenguaje utilizado en el ámbito financiero.

El avance de las técnicas de Procesamiento del Lenguaje Natural (NLP) ha abierto nuevas oportunidades para transformar el análisis financiero tradicional, permitiendo detectar patrones ocultos en los textos y evaluar su impacto en los mercados. Modelos como BERT y FinBERT han demostrado ser eficaces en el análisis de sentimiento financiero y la detección de tendencias en noticias económicas. Sin embargo, a pesar de estos avances, sigue existiendo una brecha entre la información cualitativa derivada del análisis de texto y su integración con estrategias cuantitativas utilizadas en la predicción del mercado.

Este proyecto nace de la idea de combinar el análisis de sentimiento con técnicas avanzadas de predicción de precios con el objetivo de ofrecer una visión más completa del comportamiento del mercado. La posibilidad de integrar modelos de Deep Learning con análisis de datos financieros estructurados permitirá no solo interpretar la información disponible, sino también anticipar cómo ciertos eventos o tendencias pueden afectar la evolución de los precios de los activos. Además, la implementación de una plataforma interactiva facilitará la visualización de estos resultados, proporcionando una herramienta

accesible y eficaz para analistas financieros, inversores y empresas que buscan mejorar su capacidad de toma de decisiones basada en datos.

OBJETIVOS

El proyecto tiene como objetivo principal desarrollar un sistema avanzado de análisis financiero que combine técnicas de procesamiento de lenguaje natural (NLP) y aprendizaje profundo (Deep Learning), con el fin de estudiar el impacto de las noticias financieras en el comportamiento del mercado.

Para alcanzar este objetivo, se plantean los siguientes objetivos específicos:

1. **Analizar el sentimiento de noticias financieras** mediante modelos avanzados de NLP como BERT, RoBERTa y FinBERT, clasificando el contenido en positivo, negativo o neutral.
2. **Diseñar un sistema de generación automática de informes** que resuma los resultados del análisis de sentimiento y permita la exploración por períodos temporales y compañías específicas.
3. **Integrar los resultados del análisis de sentimientos con datos financieros estructurados**, como precios de cierre y volúmenes de transacción, con el objetivo de construir modelos predictivos del comportamiento del mercado.
4. **Desarrollar una plataforma interactiva** que permita visualizar de forma clara los resultados del análisis, las predicciones y las simulaciones generadas por el sistema.

De esta forma, el proyecto busca combinar técnicas de NLP, modelado predictivo y simulación para ofrecer una solución integral que permita evaluar el impacto de las noticias financieras en los mercados de forma eficiente, accesible y orientada a la toma de decisiones.

Además de los objetivos técnicos, este proyecto busca alinearse con los Objetivos de Desarrollo Sostenible (ODS), principalmente en las áreas de industria, educación e innovación. Se ha incluido una descripción detallada de esta alineación en el Anexo A: ODS.

METODOLOGÍA

Con el fin de alcanzar los objetivos definidos previamente, se ha diseñado una metodología dividida en cinco fases. Cada fase cubre una necesidad concreta del sistema y contribuye de forma progresiva a la creación de una solución integral para el análisis y predicción del comportamiento del mercado:

1. **Recolección y preprocesamiento de datos:** La primera fase consiste en la recopilación de un corpus de noticias financieras y la obtención de datos históricos del mercado a través de la API de Yahoo Finance. El preprocesamiento textual incluye limpieza, tokenización, lematización y eliminación de *stopwords*. Este proceso garantiza la calidad y consistencia de los datos.
2. **Análisis de sentimientos:** Se utilizarán enfoques iniciales basados en diccionarios financieros (Loughran-McDonald, VADER) para establecer una referencia. A su vez, se implementarán modelos de NLP de tipo transformer, como BERT, FinBERT o RoBERTa, entrenados o ajustados específicamente para el análisis de noticias financieras. Se evaluará su rendimiento mediante métricas clásicas de clasificación, seleccionando el modelo óptimo para su integración en las etapas siguientes.
3. **Generación de informes y extracción de *insights*:** A partir de los resultados obtenidos, se desarrollará un sistema automatizado de generación de informes. Estos incluirán resúmenes de las noticias, palabras clave, análisis de tópicos y visualizaciones, permitiendo una interpretación fácil por parte del usuario. El sistema también contemplará filtros por empresa, período o tipo de sentimiento.
4. **Desarrollo de la plataforma interactiva:** Finalmente, se implementará una interfaz visual que permita al usuario explorar los resultados, predicciones y simulaciones generadas. Se priorizará una experiencia visual clara, con herramientas interactivas como filtros, comparativas y *dashboards* personalizados.

Se ha elegido el lenguaje de programación Python y entornos como Jupyter Notebook y VSCode. Bibliotecas especializadas como Pandas, Transformers de Hugging Face, NLTK, Plotly y Streamlit soportarán el procesamiento, análisis y visualización de los datos [24].

HERRAMIENTAS UTILIZADAS

Se van a emplear Jupyter Notebook y Visual Studio Code (VSCode) como los editores de código principales. El primero será clave para el análisis exploratorio de datos, el diseño y evaluación de modelos de Machine Learning (como BERT o FinBERT) [25]. Gracias a su enfoque interactivo, permite combinar código, visualizaciones y explicaciones en un único documento, lo que facilita el desarrollo del proyecto. Por su parte, VSCode será empleado para la implementación de la interfaz interactiva, la gestión del código del proyecto mediante Git y la ejecución de tareas en su terminal integrada, agilizando el desarrollo de la aplicación final [26].

En cuanto a bibliotecas de Python, Pandas desempeñará un papel fundamental en la carga, limpieza y manipulación del dataset de noticias financieras, ofreciendo estructuras de datos eficientes y herramientas para manejar grandes volúmenes de información [27][28]. Hugging Face Transformers permitirá tener acceso a modelos preentrenados de lenguaje natural como BERT y FinBERT, facilitando el análisis de sentimientos en noticias financieras al identificar el tono emocional de los textos [29][30]. Además, NLTK (Natural Language Toolkit) permite trabajar con herramientas avanzadas de procesamiento de lenguaje, como la tokenización, la lematización y el análisis de sentimientos, complementando las capacidades de los modelos [31].

Para la presentación y visualización de los resultados, se emplearán Plotly, Dash y Streamlit, bibliotecas diseñadas para crear gráficos interactivos y *dashboards* dinámicos. Estas herramientas permitirán explorar patrones, simulaciones y predicciones de mercado de forma visual e intuitiva [32][33].

ESTRUCTURA DEL TRABAJO Y PLANIFICACIÓN

El presente trabajo se estructura en torno a tres bloques principales que guían el desarrollo técnico y metodológico. En primer lugar, se realizará una exploración y comparación de distintos enfoques para el análisis de sentimientos en textos financieros, incluyendo métodos basados en diccionarios, modelos de Machine Learning tradicionales y modelos de lenguaje preentrenados como FinBERT o versiones adaptadas de transformers. El objetivo es seleccionar el modelo que mejor se ajuste al dominio financiero, tanto en precisión como capacidad de generalización. A continuación, se desarrollará un sistema automatizado de generación de informes, que extraerá y organizará los principales *insights* detectados a partir del análisis textual. Estos resultados se integrarán en una plataforma interactiva, que permitirá visualizar y consultar los análisis de forma dinámica por empresa y por período temporal.

Se muestra el plan de trabajo orientativo para el desarrollo del trabajo. La planificación del trabajo se ha dividido en tres fases secuenciales, aunque con ciertas tareas en paralelo: exploración y evaluación de modelos, desarrollo de la herramienta de análisis y visualización e integración con modelos predictivos, concluyendo con la validación de resultados y redacción final.

TAREA	ENER	FEBRERO	MARZO	ABRIL	MAY	JUNIO
Definición y Planificación						
Obtención y Preparación de datos						
Análisis de Sentimientos						
Generación de Informes						
Validación de los Modelos						
Desarrollo de la Plataforma						
Documentación						

Table 1: Planificación temporal del Trabajo Fin de Máster

Capítulo 3. OBTENCIÓN DE LOS DATOS

Para el desarrollo del trabajo ha sido necesario recopilar distintos conjuntos de datos que permitan realizar tanto el análisis de sentimientos como el desarrollo de la aplicación sobre noticias financieras reales. En esta sección se describen las fuentes de datos utilizadas, su estructura y su propósito dentro del proyecto. Se han priorizado datasets de libre acceso, con suficiente volumen y calidad para garantizar la validez del análisis.

NOTICIAS

Con el objetivo de llevar a cabo el análisis de sentimientos sobre titulares financieros y su posterior integración con datos bursátiles, se han empleado dos fuentes de datos principales, ambas obtenidas de la plataforma Kaggle [\[34\]](#).

DATASET CON ETIQUETAS DE SENTIMIENTO

La primera parte del proyecto se centra en identificar cual es el mejor modelo para el análisis de sentimientos, el cual se va a determinar mediante la comparación de distintos enfoques aplicados a titulares financieros. Para ello se ha utilizado el dataset *FinancialPhraseBank* [\[35\]](#), el cual contiene aproximadamente 71.150 registros con tres columnas: *Index*, *Sentiment* y *Sentence*. Los titulares han sido previamente categorizados en positivo, negativo o neutro desde la perspectiva de un inversor minorista.

<i>Campo</i>	<i>Tipo de dato</i>	<i>Descripción</i>
Index	Entero	Identificador único del registro dentro del dataset.
Sentiment	Texto (string)	Valor categórico que indica el sentimiento asociado al titular. Puede ser: <i>positive</i> , <i>negative</i> , o <i>neutral</i> .
Sentence	Texto (string)	Titular de noticia financiera previamente categorizado por expertos.

Table 2: Dataset con etiquetas de sentimiento

El uso de este dataset ha resultado muy útil, ya que se ha usado para la evaluación de los modelos tanto preentrenados como modelos propios con división *train-test*, sin necesidad de recurrir a técnicas auxiliares de anotación o clasificación manual. Aunque no contiene información sobre empresas o fechas, su volumen y etiquetado directo lo hacen especialmente útil para validar la capacidad de los modelos e identificar el tono predominante en noticias financieras [36].

DATASET PRINCIPAL: TITULARES CON INFORMACIÓN DE MERCADO

Para el desarrollo principal del proyecto se ha empleado un segundo dataset, que contiene titulares de prensa financiera junto con información relevante, como la fecha de publicación y el nombre del stock mencionado en el artículo [37]. A diferencia del anterior, este conjunto de datos no contiene etiquetas de sentimiento, lo que permite abordar el objetivo principal del proyecto, la clasificación automática del sentimiento asociado a cada titular.

El dataset fue obtenido a través de una recopilación manual y *scripts* de *scraping*, puesto que acceder a bases de datos históricas de noticias financieras suele implicar costes elevados. La información cubre el periodo 2009–2020 y está publicada bajo licencia CC0 (dominio público), por lo que su uso es completamente abierto y gratuito.

<i>Campo</i>	<i>Tipo de dato</i>	<i>Descripción</i>
date	Fecha	Fecha de publicación del titular (YYYY-MM-DD).
stock	Texto (string)	<i>Ticker</i> bursátil asociado al titular (ej. AAPL, MSFT).
headline	Texto (string)	Titular de prensa financiera sin etiquetar.
source (opcional)	Texto (string)	Medio o plataforma de origen del titular, si está disponible.

Table 3: Estructura del dataset de titulares financieros

Este dataset representa la base del trabajo, ya que permite analizar la evolución de los titulares en relación con el mercado financiero real. Al carecer de etiquetado, su uso está ligado directamente a la aplicación de los modelos tratados durante el proyecto.

Además, con el objetivo de facilitar la interpretación de los resultados y mejorar la claridad en las visualizaciones y en el desarrollo de la aplicación, se han utilizado documentos auxiliares que relacionan cada *ticker* bursátil (símbolo del *stock*) con el nombre completo de la empresa correspondiente. De esta forma, se consiguen asociar los símbolos, por ejemplo, AAPL, por los nombres de las empresas correspondientes, en este caso *Apple Inc.*, facilitando la comprensión del análisis [38].

STOCK

Continuando con el desarrollo del proyecto, se requería incorporar información financiera real de mercado que permita contextualizar y enriquecer el análisis. En este apartado se describe el proceso seguido para la obtención de los datos bursátiles asociados a las compañías mencionadas en el dataset principal.

La extracción de los datos históricos se ha realizado utilizando la librería **yfinance** [39] que permite acceder de forma sencilla a la API pública de Yahoo Finance. Aunque inicialmente se consideraron otras alternativas, ésta resultó ser la opción más adecuada por varias razones:

- Permite especificar intervalos temporales exactos.
- Ofrece gran cobertura de compañías internacionales.
- Tiene un uso simple y directo en Python.
- La documentación y estructura de las respuestas es clara y fácilmente integrable.

Yahoo Finance es una plataforma ampliamente utilizada en el ámbito financiero que proporciona datos históricos y en tiempo real sobre mercados bursátiles, incluyendo precios de apertura y cierre, máximos, mínimos y volúmenes de transacción. Su API pública no oficial, accesible a través de **yfinance**, ha sido adoptada para realizar estudios basados en series temporales financieras [40].

Para construir el dataset, se partió de los *tickers* de las compañías recogidas en el dataset principal de titulares, ya que son las entidades de interés para el proyecto. Se identificaron

mediante código Python las fechas mínima y máxima presentes en el conjunto de datos, confirmando que cubrían el periodo de 2009 a 2020. Con esta información se automatizó la descarga de las series temporales correspondientes desde Yahoo Finance.

Inicialmente, los datos de cada compañía fueron almacenados en archivos CSV individuales. Una vez finalizado el proceso de recopilación, todos los archivos fueron combinados en un único dataset, lo que facilita su análisis conjunto y su integración con los titulares de prensa ya procesados. El diccionario de datos del archivo final resultó en:

<i>Campo</i>	<i>Tipo de dato</i>	<i>Descripción</i>
Company	Texto (string)	<i>Ticker</i> bursátil de la compañía (ej. AAPL, MSFT).
Date	Fecha	Fecha del registro (formato YYYY-MM-DD).
Open	Decimal (float)	Precio de apertura de la acción en esa fecha.
High	Decimal (float)	Precio máximo alcanzado durante la jornada.
Low	Decimal (float)	Precio mínimo registrado en la jornada.
Close	Decimal (float)	Precio de cierre de la acción en esa fecha.
Volume	Entero	Número de acciones intercambiadas durante la jornada.

Table 4: Estructura del dataset histórico de precios bursátiles

Para una descripción detallada de la metodología empleada en la extracción de los datos financieros, incluyendo el código utilizado, puede consultarse el Anexo B: Extracción de Datos Financieros.

PREPROCESAMIENTO DE DATOS

Una vez obtenidos los datos necesarios para el desarrollo del proyecto, se llevó a cabo un proceso de preprocesamiento tanto sobre los titulares de noticias financieras como sobre los datos bursátiles. Esta etapa es crítica, ya que garantiza la calidad de los datos y permite que los modelos de Machine Learning puedan operar de forma eficiente, reduciendo el riesgo de errores y mejorando la precisión de los resultados.

En primer lugar, se realizaron tareas de limpieza y transformación sobre los conjuntos de datos relacionados con los titulares de las noticias financieras. Para el dataset etiquetado (*sentiment_stock_data.csv*), se estandarizaron los nombres de las columnas y se eliminaron aquellas que no aportaban valor al análisis (por ejemplo, *Unnamed: 0*). Asimismo, se descartaron registros con valores nulos o mal formateados. En el caso del dataset sin etiquetar (*analyst_ratings_processed.csv*), se realizó una limpieza exhaustiva de la columna *date*, eliminando filas con fechas ausentes y transformando el campo a un formato homogéneo (YYYY-MM-DD), asegurando además la consistencia temporal mediante la conversión explícita a zona horaria UTC. Este paso es especialmente importante para facilitar la alineación temporal con los datos del mercado extraídos de Yahoo Finance.

Otro aspecto clave fue la preparación de las etiquetas de sentimiento. En el dataset etiquetado, las clases *positive*, *neutral* y *negative* se transformaron a valores numéricos, permitiendo su integración en modelos supervisados que requieren representaciones numéricas de las variables categóricas. Esta conversión no solo facilita el entrenamiento del modelo, sino que permite también calcular métricas de rendimiento estándar en clasificación, como *precision*, *recall* o *F1-score*. Se analizó la distribución de las clases en el conjunto de datos etiquetados con el objetivo de detectar posibles desequilibrios, aplicando técnicas de submuestreo.

Dado que la distribución de clases en el dataset original era significativamente desequilibrada (*neutral*: 2878; *negative*: 604; *positive*: 1363), se aplicó una técnica de

submuestreo (*undersampling*) para reducir el número de ejemplos de la clase mayoritaria y equilibrar así el conjunto de datos. Esta técnica ayuda a evitar que el modelo se sesgue hacia la clase más frecuente, mejorando su capacidad para generalizar y detectar los casos menos representados (por ejemplo, titulares negativos). Aunque el submuestreo implica perder parte de la información original, en este caso fue una decisión adecuada dado el tamaño considerable del conjunto de datos inicial.

En cuanto a los datos financieros, el preprocesamiento se centró en asegurar la calidad de los valores numéricos correspondientes a los precios (*Open, High, Low, Close*) y al volumen de transacciones (*Volume*). Se verificó la consistencia de las fechas y se aplicaron transformaciones similares a las del dataset de titulares para unificar el formato temporal.

Este proceso de preprocesamiento ha sido esencial para garantizar la calidad, coherencia y relevancia de los datos que serán empleados en la evaluación de los modelos estudiados en este trabajo y el diseño de la plataforma final.

Capítulo 4. ANÁLISIS DE SENTIMIENTOS

El análisis de sentimientos es una técnica del procesamiento del lenguaje natural (NLP) que permite identificar, extraer y clasificar automáticamente las emociones expresadas en un texto. En el contexto financiero, esta disciplina resulta especialmente útil para evaluar el tono de noticias, informes o publicaciones en redes sociales, con el objetivo de anticipar el comportamiento del mercado ante determinados eventos [7].

La relación entre el sentimiento y la evolución del mercado financiero ha sido ampliamente estudiada. Aunque está demostrado que las noticias con un tono muy positivo pueden influir en la percepción de los inversores y favorecer movimientos alcistas, también es habitual que los medios reflejen o incluso amplifiquen tendencias del mercado ya en marcha. Con frecuencia, se crea una retroalimentación entre el sentimiento expresado en los textos y la dinámica de los mercados, por lo que resulta especialmente valioso contar con herramientas capaces de analizar el tono general de la información financiera.

En este capítulo se va a exponer la implementación y el estudio progresivo de varios enfoques de análisis de sentimientos aplicados al dominio financiero, ordenados según su complejidad técnica. Se va a empezar por métodos clásicos basados en diccionarios, seguidos de modelos de redes neuronales como LSTM y, finalmente, se analizarán modelos preentrenados basados en arquitecturas Transformer, como BERT y FinBERT. Cada uno de los métodos se ha implementado, evaluado y justificado en función de su rendimiento, mostrando en cada caso los motivos que han llevado continuar con el siguiente enfoque más sofisticado.

Para garantizar la calidad del análisis de sentimientos, se ha implementado un *pipeline* de preprocesamiento que permite limpiar y estandarizar los titulares financieros antes de ser evaluados por los modelos. En primer lugar, se aplica una función de normalización que convierte el texto a minúsculas, elimina URLs y mantiene únicamente caracteres relevantes (letras, números y signos con carga semántica como %, \$, ! y ?). A continuación, se lleva a

cabo una tokenización robusta utilizando expresiones regulares, lo que permite separar adecuadamente palabras y números. Se eliminan las *stopwords* genéricas y se aplica lematización para reducir las palabras a su forma base, con el objetivo de maximizar la consistencia en el análisis. Este proceso asegura una representación textual clara y uniforme, imprescindible para modelos que dependen del contenido léxico para inferir el sentimiento.

Para la evaluación de los distintos enfoques de análisis de sentimiento, se emplearán diversas métricas de rendimiento, las cuales se describen en el Anexo C: Métricas de Evaluación.

ENFOQUE BASADO EN DICCIONARIOS

Los enfoques basados en diccionarios se caracterizan por ser una de las metodologías más simples y tradicionales para el análisis de sentimientos. Se basan en listas predefinidas de palabras asociadas a una polaridad concreta (positiva o negativa), y su funcionamiento consiste en contar la frecuencia de dichas palabras en un texto para inferir el sentimiento global. Su principal ventaja es la interpretabilidad, dado que es fácil entender por qué una noticia ha sido clasificada de una manera u otra. No requieren entrenamiento previo, lo que los convierte en una opción accesible y rápida de aplicar. Sin embargo, presentan limitaciones importantes, como su incapacidad para captar el contexto, las negaciones o la ironía [\[41\]](#).

LOUGHRAN-McDONALD

El diccionario de Loughran y McDonald se diseñó específicamente para el análisis de textos financieros [\[42\]](#). A diferencia de otros diccionarios de sentimientos como Harvard IV o LIWC, Loughran-McDonald está construido a partir de miles de documentos regulatorios e informes anuales, lo que le permite capturar con mayor precisión el lenguaje técnico y típico del ámbito económico.

Se caracteriza por etiquetar palabras en diferentes categorías relevantes para el análisis financiero, tales como *positive*, *negative*, *uncertainty*, *litigious*, entre otras. Esto permite una interpretación del tono más alineada con la semántica del sector [43].

Dado que se está trabajando con titulares financieros y se pretende evaluar la polaridad de textos breves, el uso de Loughran-McDonald representa una opción lógica como punto de partida. Lo que se pretende es establecer una línea base con una implementación interpretable, rápida y correcta, sobre la cual se compararán después enfoques más complejos, como redes neuronales o modelos Transformer.

Inicialmente, se optó por una aproximación sencilla que consideraba únicamente el número de palabras positivas y negativas encontradas en cada titular, calculando un primer *sentiment score* que consistía en la diferencia entre la cantidad de veces que se repetía ambas etiquetas.

	Caso	Interpretación	Ejemplo
Valor Positivo	$n_{pos} > n_{neg}$	Sentimiento positivo	3 palabras positivas y 1 negativa Score = 2 \equiv Positivo
Valor negativo	$n_{pos} < n_{neg}$	Sentimiento negativo	1 palabra positiva y 3 negativas Score = -2 \equiv Negativo
Valor cercano a cero	$n_{pos} \approx n_{neg}$	Sentimiento neutro	2 palabras positivas y 2 negativas Score = 0 \equiv Neutro

Table 5: Interpretación del *sentiment score* básico

Posteriormente, se amplió la lógica incorporando dos categorías adicionales presentes en el diccionario: *litigious* y *uncertainty*. Ambas son frecuentes en titulares financieros y, aunque no siempre implican un sentimiento negativo, sí representan una connotación conflictiva, como pueden ser *lawsuit* (demanda), *penalty*, *fine*, *sanction* (multa, sanción), o de ambigüedad. Para tener en cuenta su efecto, se ha diseñado un *sentiment score* ajustado que penaliza la presencia de estos términos mediante la aplicación de pesos negativos.

	sentiment	title_clean	label_Sentiment_Score	label_Sentiment_Score_Adjust	n_pos	n_neg	n_unc	n_lit
1	negative	international electronic industry company elco...	negative	negative	0	2	0	1
414	negative	tinyurl link take user scamming site promising...	neutral	neutral	0	0	0	0
420	negative	compared ftse 100 index rose 36 7 point 0 6 da...	neutral	neutral	0	0	0	0
422	negative	compared ftse 100 index rose 94 9 point 1 6 da...	neutral	neutral	0	0	0	0
499	negative	one challenge oil production north sea scale f...	negative	negative	0	2	0	0

Ilustración 1: Aplicación del diccionario Loughran-McDonald

La puntuación final se ha normalizado, dividiéndola por el número total de palabras detectadas en todas las categorías, para evitar que titulares más largos acumularan más puntuación por el simple hecho de contener más palabras. De este modo, se garantiza una evaluación correcta del tono, permitiendo comparar de forma justa titulares de distinta longitud.

Se evaluaron distintas combinaciones de pesos para *uncertainty* y *litigious* (0.25, 0.5, 0.75 y 1.0), aplicando para cada una un *grid search* sobre el umbral de clasificación. Se seleccionó en cada caso el umbral que maximizaba la entropía de la distribución de clases. Este criterio permitió identificar el umbral que generaba una clasificación más equilibrada entre etiquetas positivas, negativas y neutrales. Tras evaluar todos los valores de umbral definidos en el *grid* (0.1 a 0.95, con incrementos de 0.05), se observó que en todos los casos la entropía de la distribución de clases se maximizaba en el valor de ± 0.1 :

Mejores umbrales por combinación de pesos:

```
{'score_u25_125': np.float64(0.1), 'score_u50_150': np.float64(0.1), 'score_u75_175': np.float64(0.1), 'score_u100_1100': np.float64(0.1)}
```

Por este motivo, se seleccionó ± 0.1 como umbral de corte final para todas las variantes del modelo. Tras comparar cada variante mediante las métricas de *precision*, *recall* y *F1-score*:

Resultados para score_u25_l25 (umbral = 0.1):					Resultados para score_u75_l75 (umbral = 0.1):				
	precision	recall	f1-score	support		precision	recall	f1-score	support
negative	0.56	0.47	0.51	604	negative	0.56	0.47	0.51	604
neutral	0.40	0.73	0.51	604	neutral	0.40	0.73	0.51	604
positive	0.71	0.24	0.36	604	positive	0.71	0.24	0.36	604
accuracy			0.48	1812	accuracy			0.48	1812
macro avg	0.56	0.48	0.46	1812	macro avg	0.56	0.48	0.46	1812
weighted avg	0.56	0.48	0.46	1812	weighted avg	0.56	0.48	0.46	1812

Resultados para score_u50_l50 (umbral = 0.1):					Resultados para score_u100_l100 (umbral = 0.1):				
	precision	recall	f1-score	support		precision	recall	f1-score	support
negative	0.56	0.47	0.51	604	negative	0.56	0.47	0.51	604
neutral	0.40	0.73	0.51	604	neutral	0.40	0.73	0.51	604
positive	0.71	0.24	0.36	604	positive	0.71	0.23	0.35	604
accuracy			0.48	1812	accuracy			0.48	1812
macro avg	0.56	0.48	0.46	1812	macro avg	0.56	0.48	0.46	1812
weighted avg	0.56	0.48	0.46	1812	weighted avg	0.56	0.48	0.46	1812

Ilustración 2: Comparativa de resultados para diferentes variantes de sentiment score con umbral ±0.1.

Se observó que los resultados obtenidos eran prácticamente idénticos, independientemente del peso. Esto sugiere que el enfoque basado en diccionario es relativamente estable ante pequeños ajustes. Por ello, se decidió mantener la configuración estándar (0.5, 0.5).

Una vez aplicados y evaluados tanto el *sentiment score* básico (basado únicamente en la diferencia neta entre palabras positivas y negativas) como la versión ajustada (que incluía penalizaciones para términos categorizados como *uncertainty* o *litigious*), se observó que el enfoque básico logra una mayor precisión global, especialmente en la clase negative (0.70 vs. 0.56). Sin embargo, presenta un fuerte desbalance entre clases, con un *recall* muy bajo para *negative* (0.43) y *positive* (0.24), y un exceso de ejemplos clasificados como neutral. Por el contrario, el score ajustado consigue una distribución de etiquetas más equilibrada, reflejada en un *F1-score* más homogéneo entre clases, a costa de perder precisión en ciertas categorías.

Score básico:	precision	recall	f1-score	support	Score ajustado:	precision	recall	f1-score	support
	negative	0.70	0.43	0.53		604	negative	0.56	0.47
neutral	0.41	0.83	0.55	604	neutral	0.40	0.73	0.51	604
positive	0.71	0.24	0.36	604	positive	0.71	0.24	0.36	604
accuracy			0.50	1812	accuracy			0.48	1812
macro avg	0.61	0.50	0.48	1812	macro avg	0.56	0.48	0.46	1812
weighted avg	0.61	0.50	0.48	1812	weighted avg	0.56	0.48	0.46	1812

Ilustración 3: Evaluación del score básico vs. ajustado mediante métricas de clasificación por clase.

Estos resultados sugieren que la versión ajustada ofrece una representación más realista y prudente del tono del titular, siendo más adecuada como punto de partida.

Sin embargo, este enfoque presenta limitaciones importantes. La más significativa es su incapacidad para capturar el contexto semántico de las palabras, lo que afecta directamente a la precisión del análisis. Al depender únicamente de la presencia de términos etiquetados como positivos, negativos, de incertidumbre o litigiosos, el modelo no puede interpretar frases donde el significado depende de la estructura sintáctica o del uso del lenguaje, como ocurre con negaciones, expresiones irónicas o sarcasmo.

Además, el desequilibrio inherente en el diccionario, con una cantidad considerablemente mayor de términos negativos frente a positivos, tiende a generar una clasificación sesgada, favoreciendo la asignación de etiquetas *negative* o *neutral* en exceso. Esto se refleja en las métricas obtenidas, donde la única clase con una *precision* y *recall* relativamente aceptables es la *neutral*, mientras que *positive* es la categoría que más errores de clasificación acumula.

VADER

VADER (*Valence Aware Dictionary and sEntiment Reasoner*) es un modelo de análisis de sentimientos basado en un diccionario optimizado para textos breves e informales, como titulares, tweets o reseñas. Fue desarrollado por C.J. Hutto y Eric Gilbert (2014) y forma parte de la librería `nltk` [\[44\]](#).

A diferencia de los enfoques tradicionales que solo consideran palabras positivas o negativas, VADER añade puntuaciones de intensidad semántica y un conjunto de reglas para

interpretar negaciones, emoticonos, adverbios intensificadores (por ejemplo, distinguir “very good” de “good”) y signos de puntuación (como “!!!”). Gracias a esta lógica basada en reglas (*rule-based*), resulta un punto intermedio entre un lexicón puro (como Loughran-McDonald) y un modelo de aprendizaje automático.

Para implementar VADER, se utilizó el analizador de sentimientos preentrenado *SentimentIntensityAnalyzer* de la librería *nltk*. La clasificación se aplicó directamente sobre los titulares del dataset. Se definió una función que calculaba una puntuación compuesta (*compound score*) para cada texto, con valores entre -1 (muy negativo) y +1 (muy positivo), identificando así el tono emocional general del titular. La clasificación final se realizó asignando etiquetas *positive*, *negative* o *neutral* según los umbrales estándar de ± 0.1 .

Además, se definió una función de limpieza previa que convierte el texto a minúsculas, elimina puntuación, números y palabras vacías (*stopwords*), manteniendo expresamente términos relevantes como “no”, “not” y “nor” para mantener el sentido negativo en ciertas frases. Aunque VADER puede aplicarse directamente sobre texto sin procesar, esta fase de limpieza se mantuvo para garantizar la consistencia metodológica con los demás enfoques del benchmark. La implementación final consistió en aplicar esta lógica a cada titular y asignar automáticamente la etiqueta de sentimiento correspondiente.

COMPARACIÓN DE LAS MÉTRICAS DE RENDIMIENTO

Al comparar los resultados obtenidos con los tres enfoques basados en diccionario (Loughran-McDonald básico, Loughran-McDonald ajustado y VADER) se aprecian diferencias relevantes tanto en el comportamiento por clase como en el rendimiento general.

	sentiment	title_clean	lm_sentiment	lm_sentiment_adj	vader_sentiment
1	negative	international electronic industry company elco...	negative	negative	neutral
414	negative	tinyurl link takes users scamming site promisi...	neutral	neutral	positive
420	negative	compared ftse 100 index rose 36 7 points 0 6 %...	neutral	neutral	neutral
422	negative	compared ftse 100 index rose 94 9 points 1 6 %...	neutral	neutral	neutral
499	negative	one challenges oil production north sea scale ...	negative	negative	neutral
538	negative	jan 6 ford struggling face slowing truck suv s...	negative	negative	negative
541	negative	peer peugeot fell 0 81 pct sales rose 6 3 pct ...	neutral	neutral	neutral

Ilustración 4: Comparación de los resultados obtenidos por las técnicas basadas en diccionarios

A nivel general, VADER ofrece un rendimiento ligeramente superior al ajustado de Loughran, y comparable al modelo básico en relación al *accuracy* y *F1* global. El ajuste de pesos en Loughran no mejora los resultados respecto al modelo base, aunque sí balancea un poco la distribución de clases (*macro avg* más coherente).

Modelo	Accuracy	Macro F1	Weighted F1
Loughran-McDonald (básico)	0.50	0.48	0.48
Loughran-McDonald (ajustado)	0.48	0.46	0.46
VADER	0.50	0.49	0.49

Table 6: Comparación global del rendimiento entre los modelos basados en diccionario

Loughran-McDonald básico alcanza una precisión destacable en las clases *negative* y *positive*, pero muestra una fuerte tendencia a clasificar como *neutral*, lo que se traduce en un *recall* excesivamente alto para esa clase y un rendimiento desequilibrado. La versión ajustada, con penalización sobre términos de *uncertainty* y *litigious*, no logra mejorar las métricas, aunque ofrece una ligera redistribución de las etiquetas que equilibra mejor el *macro F1-score*.

En la **clase *negative***, el enfoque Loughran-McDonald básico obtiene el mejor equilibrio entre *precisión* y *recall*, aunque el ajuste posterior reduce esa precisión sin una mejora clara.

VADER, por su parte, presenta la mayor precisión en esta clase, pero con un *recall* más bajo, lo que indica que acierta cuando clasifica como negativo, pero lo hace con menor frecuencia.

Modelo	Precision	Recall	F1-Score
LM básico	0.70	0.42	0.53
LM ajustado	0.56	0.46	0.51
VADER	0.73	0.29	0.42

Table 7: Comparativa del rendimiento por clase negativa

Respecto a la **clase *neutral***, Loughran-McDonald básico ofrece el mayor *recall* (0.84), aunque a costa de cometer errores significativos en las otras dos clases. VADER, en cambio, mejora en precisión y muestra un comportamiento más balanceado.

Modelo	Precision	Recall	F1-Score
LM básico	0.41	0.84	0.55
LM ajustado	0.40	0.73	0.51
VADER	0.47	0.53	0.50

Table 8: Comparativa del rendimiento por clase neutral

Es en la **clase *positive*** donde VADER se diferencia de forma clara, alcanzando un *recall* del 0.69 frente al 0.24 de los dos modelos Loughran-McDonald. Aunque su precisión en esta clase es más baja, logra el mejor *F1-score*, mostrando su mayor capacidad para captar sentimientos positivos incluso en expresiones menos directas.

Modelo	Precision	Recall	F1-Score
LM básico	0.72	0.24	0.36
LM ajustado	0.72	0.24	0.36
VADER	0.47	0.69	0.56

Table 9: Comparativa del rendimiento por clase *positive*

En conjunto, VADER presenta un rendimiento más homogéneo, con mejores resultados en dos de las tres clases y una clasificación menos sesgada.

MODELOS LSTM

Tras explorar enfoques no supervisados basados en diccionarios, como Loughran-McDonald y VADER, se ha observado que, si bien ofrecen interpretabilidad y bajo coste computacional, presentan limitaciones importantes. Su incapacidad para capturar el contexto semántico, la estructura gramatical o la interacción entre palabras impide que comprendan frases complejas, negaciones o información implícita en el lenguaje financiero.

Con el objetivo de realizar un análisis exhaustivo, se va a optar como siguiente paso por el uso de modelos supervisados basados en redes neuronales recurrentes, en particular las arquitecturas LSTM (Long Short-Term Memory) y GRU (Gated Recurrent Unit). Estos modelos están específicamente diseñados para trabajar con secuencias de texto, ya que incorporan una memoria interna que permite retener información de palabras anteriores.

A diferencia de los enfoques anteriores, las LSTM y GRU no tratan cada palabra por separado, sino que tiene en cuenta las palabras que la rodean (antes o después) para entender su verdadero significado. Son capaces de modelar la dependencia contextual entre palabras, lo cual es especialmente relevante en titulares financieros, donde el significado de una expresión puede variar radicalmente en función de su entorno léxico [\[45\]](#).

LSTM BÁSICO

La primera arquitectura implementada fue una red LSTM básica. Se optó por una estructura bidireccional para que el modelo pudiera capturar tanto el contexto previo como posterior a cada palabra dentro de un titular, lo cual es especialmente útil en textos breves donde cada término puede influir en el sentido global de la frase. Se emplearon dos capas LSTM de 64 y 32 neuronas respectivamente. La primera proporciona una gran capacidad de representación, mientras que la segunda consolida abstracciones más elevadas, es decir, transforma y resume la información que ya ha sido procesada por la capa anterior, lo que facilita la extracción progresiva de patrones semánticos.

Las capas LSTM se intercalaron con capas de *dropout* (tasa 0.4) y *regularización L2* para reducir el riesgo de sobreajuste, dado el tamaño limitado del conjunto de datos. Se utilizó como función de pérdida la *categorical crossentropy*, ya que se trata de una tarea de clasificación multiclase con etiquetas codificadas en *one-hot* (*negative: 0; neutral: 1; positive: 2*). El optimizador elegido fue Adam, por su eficacia y estabilidad en tareas de procesamiento de lenguaje natural. Finalmente, se incorporó una estrategia de *early stopping* basada en la pérdida de validación para detener el entrenamiento cuando no se observaban mejoras, evitando así entrenamientos innecesarios y reduciendo la posibilidad de sobreentrenamiento.

Gracias a estos ajustes el modelo LSTM logró una precisión global (*accuracy*) del 52.8%, mejorando ligeramente respecto a los enfoques no supervisados y a versiones iniciales, donde no se había incluido el bidireccionamiento o la función de pérdida la *categorical crossentropy*. Las métricas por clase muestran un rendimiento razonablemente equilibrado, aunque aún con margen de mejora, especialmente en la clase *positive* (2):

	Precision	Recall	F1-score	Support
0	0.55	0.75	0.64	128
1	0.64	0.47	0.54	107
2	0.41	0.36	0.38	128

Table 10: Métricas de Rendimiento LSTM Bidireccional

A nivel de clases, el mejor rendimiento se observó en la categoría negativa, con una precisión del 55 % y una *recall* del 75 %, lo que indica que el modelo identifica correctamente la mayoría de los titulares negativos. En cambio, el rendimiento fue más limitado en la clase neutral ($F1\text{-score} = 0.50$) y especialmente en la clase positiva, donde tanto la precisión (0.41) como la *recall* (0.36) se vieron afectadas, probablemente porque los vectores (*embeddings*) que la red aprendió “desde cero” no contienen suficiente información semántica. Aun así, tanto la *macro F1* como la *weighted F1* se situaron en torno al 0.53, lo que sugiere un rendimiento razonablemente equilibrado entre clases, sin un sesgo tan marcado como el

observado en modelos como VADER. Estos resultados refuerzan la utilidad de las redes recurrentes para el análisis de sentimientos, aunque también muestran la necesidad de incorporar *embeddings* preentrenados más ricos, como GloVe o FastText, para mejorar la comprensión del contexto lingüístico.

LSTM JUNTO CON GLOVE

GloVe (*Global Vectors for Word Representation*) es un método de generación de vectores de palabras preentrenados que combina información estadística de coocurrencias entre palabras en grandes corpus textuales. A diferencia de los enfoques clásicos que aprenden los embeddings durante el entrenamiento, GloVe proporciona representaciones significativas desde el inicio, capturando relaciones semánticas como similitud y analogía entre términos. Esto permite al modelo partir con un conocimiento lingüístico previo, lo que resulta especialmente útil en datasets de tamaño reducido como el presente [46]. Se utilizó el conjunto de vectores preentrenados *glove.6B.100d*, entrenado por Stanford sobre un corpus combinado de Wikipedia 2014 y Gigaword 5, con un total de aproximadamente 6.000 millones de palabras [47]. Este modelo genera representaciones de 100 dimensiones para unas 400.000 palabras. Se ha elegido, ya que ofrece un buen equilibrio entre riqueza semántica y coste computacional. El embedding se mantuvo como entrenable (`trainable=True`) para que pudiera ajustarse al dominio financiero específico durante el entrenamiento.

El modelo alcanzó una precisión en test del 70.2 %, superando destacablemente los resultados anteriores tanto de modelos supervisados como no supervisados. Las métricas por clase muestran este salto cualitativo:

	Precision	Recall	F1-score	Support
0	0.82	0.70	0.76	128
1	0.68	0.63	0.65	107
2	0.63	0.77	0.69	128

Table 11: Métricas de Rendimiento LSTM + GloVe

Se puede observar como el sentimiento negativo obtiene una precisión del 82 % y un *F1-score* de 0.76, mostrando un alto grado de fiabilidad. La clase neutral presenta un rendimiento equilibrado (*F1-score* = 0.71), mientras que el sentimiento positivo, que en versiones previas era el más difícil de captar, logra ahora una *recall* del 77 % y un *F1-score* de 0.69, lo cual indica una capacidad mucho mayor del modelo para identificar titulares con tono optimista.

Estas mejoras se explican, principalmente, a la riqueza semántica de los vectores GloVe, que aporta al modelo conocimiento lingüístico previo que no requiere ser aprendido desde cero, permitiendo así generalizar mejor incluso en un corpus de tamaño moderado. Además, la arquitectura refinada con dos capas LSTM (128 y 64 unidades), regularización mediante *dropout* y *batch normalization*, y una capa densa intermedia con activación *LeakyReLU*, ha demostrado ser efectiva en capturar tanto dependencias locales como globales en los titulares.

LSTM JUNTO CON FASTTEXT

FastText es un modelo de generación de embeddings desarrollado por Facebook AI Research que, a diferencia de GloVe o Word2Vec, representa cada palabra no como un vector único, sino como la suma de los vectores de sus subpalabras o n-gramas de caracteres. Esta característica le permite capturar de forma más precisa la estructura morfológica interna de las palabras, lo cual resulta especialmente útil en contextos como el financiero, donde abundan tecnicismos, abreviaciones y términos poco frecuentes. Gracias a esta propiedad, FastText puede generar embeddings incluso para palabras no vistas durante su entrenamiento (out-of-vocabulary, OOV), lo que incrementa significativamente la robustez del modelo en aplicaciones reales [\[48\]](#).

Se ha optado por utilizar el modelo *fasttext-wiki-news-subwords-300*, preentrenado sobre un corpus de noticias, que proporciona vectores de 300 dimensiones.

A partir del vocabulario generado mediante el tokenizer (un preprocesador que transforma texto libre en secuencias de enteros según la frecuencia de las palabras), se construyó una matriz de embeddings donde cada fila representa una palabra y su correspondiente vector FastText. En los casos en que no se encontró una representación preentrenada, se asignaron vectores aleatorios dentro de un rango controlado.

La arquitectura del modelo incluyó una capa Embedding que cargaba dicha matriz, activando la opción `trainable=True` para permitir el ajuste fino (*fine-tuning*) de los vectores al dominio específico del análisis. Se añadieron dos capas LSTM bidireccionales de 128 y 64 neuronas respectivamente, intercaladas con capas de *Batch Normalization* y *Dropout* (0.4), para estabilizar el aprendizaje y mitigar el riesgo de sobreajuste. Tras estas capas recurrentes, se introdujo una capa densa de 32 neuronas con activación *LeakyReLU*, seguida de una capa de salida *softmax* para realizar la clasificación multiclase.

El modelo se entrenó con la función de pérdida *categorical_crossentropy* y el optimizador Adam, incorporando además una estrategia de *Early Stopping*. Esta lógica de entrenamiento fue diseñada de forma análoga a la utilizada en la implementación con *GloVe*, lo que permite una comparación consistente entre enfoques.

El modelo ha alcanzado una *accuracy* del 68%, ligeramente inferior a la obtenida con *GloVe* (70%).

	Precision	Recall	F1-score	Support
0	0.77	0.67	0.72	128
1	0.60	0.65	0.63	107
2	0.67	0.71	0.69	128

Table 12: Métricas de Rendimiento LSTM + FastTest

En términos de rendimiento por clase, la etiqueta *negative* ha sido la mejor clasificada, con una *precision* del 77% y un *F1-score* de 0.72, seguida por *positive* con un *F1-score* de 0.69. La clase *neutral*, en cambio, ha resultado la más difícil de identificar, con un *F1-score* de

0.63. Aunque el rendimiento general es correcto, el modelo tiende a confundir *neutral* con *positive*, lo que sugiere que la distinción semántica entre ambas clases no está bien capturada por los embeddings. Esta ligera pérdida de rendimiento con respecto a GloVe puede deberse a varios factores. El uso de vectores de mayor dimensión (300 vs. 100) aumenta la complejidad del modelo, lo que en un conjunto de datos limitado puede inducir sobreajuste. Además, la mayor robustez de FastText frente a palabras raras no parece haber supuesto una ventaja significativa, ya que los titulares financieros utilizados en el dataset son relativamente limpios y estandarizados. Se puede concluir que, aunque el modelo logra resultados razonables, no consigue superar el rendimiento de GloVe en este caso concreto.

COMPARACIÓN DE LAS MÉTRICAS DE RENDIMIENTO

Tras la implementación de los tres modelos basados en LSTM (básico, con GloVe y con FastText), se presentan a continuación los resultados globales de rendimiento para cada uno de ellos:

Modelo	Accuracy	Macro F1	Weighted F1
LSTM Básico	0.53	0.52	0.52
LSTM + GloVe	0.70	0.70	0.70
LSTM + FastText	0.68	0.68	0.68

Table 13: Comparación global del rendimiento entre los modelos basados en LSTM

A nivel general, la inclusión de embeddings preentrenados mejora de forma significativa las métricas, especialmente en la detección del sentimiento positivo, que suele ser más difícil de identificar.

El modelo LSTM básico presenta cierta mejora respecto a los enfoques no supervisados. A pesar de su simplicidad, alcanza una precisión destacable en la clase negativa (*F1-score*: 0.64), pero muestra dificultades a la hora de clasificar correctamente los titulares positivos (*F1-score*: 0.38), reflejando la necesidad de vectores semánticos más ricos.

La incorporación de embeddings GloVe aporta una mejora notable, con una *accuracy* del 70% y métricas equilibradas en las tres clases. Destaca especialmente su capacidad para detectar titulares con tono positivo, y presenta el *F1-score* más alto en todas las clases.

En la **clase *negative***, todos los modelos presentan un rendimiento elevado, siendo el modelo con GloVe el que obtiene la mayor precisión (0.82) y *F1-score* (0.76). FastText también muestra un comportamiento robusto, aunque con una ligera caída respecto a GloVe.

Modelo	Precision	Recall	F1-Score
LSTM Básico	0.55	0.75	0.64
LSTM + GloVe	0.82	0.70	0.76
LSTM + FastText	0.77	0.67	0.72

Table 14: Comparativa del rendimiento por clase *negative*

La **clase *neutral*** representa un reto para todos los modelos, aunque tanto GloVe como FastText alcanzan resultados aceptables. El modelo básico, en cambio, tiene un *recall* muy bajo (0.47), lo que implica que no reconoce adecuadamente esta categoría.

Modelo	Precision	Recall	F1-Score
LSTM Básico	0.64	0.47	0.54
LSTM + GloVe	0.68	0.63	0.65
LSTM + FastText	0.60	0.65	0.63

Table 15: Comparativa del rendimiento por clase *neutral*

La mejora más significativa se observa en la **clase *positive***. Mientras que el LSTM básico simplemente presenta un *F1-score* del 0.38, los modelos con embeddings preentrenados elevan esta métrica por encima del 0.69, mostrando una comprensión mucho más completa del contexto semántico.

Modelo	Precision	Recall	F1-Score
LSTM Básico	0.41	0.36	0.38
LSTM + GloVe	0.63	0.77	0.69
LSTM + FastText	0.67	0.71	0.69

Table 16: Comparativa del rendimiento por clase positiva

Por tanto, el modelo más equilibrado es el resultante de combinar LSTM junto con GloVe.

BERT

BERT (Bidirectional Encoder Representations from Transformers) es un modelo de lenguaje desarrollado por Google en 2018 que revolucionó el campo del procesamiento del lenguaje natural (NLP). Su principal innovación consiste en que, a diferencia de modelos anteriores que procesaban el texto de forma unidireccional (solo de izquierda a derecha o viceversa), BERT lee el texto en ambas direcciones simultáneamente (bidireccional), lo que le permite entender mejor el contexto completo de una palabra dentro de una frase [49] [50].

Una característica clave de BERT es que es un modelo preentrenado. Esto significa que, antes de ser utilizado para una tarea específica, ha sido entrenado con grandes corpus de texto general, como noticias o libros, para aprender los patrones y estructuras del lenguaje. Se basa en la arquitectura Transformer, concretamente en su parte "encoder", y se entrena utilizando dos tareas principales:

- *Masked Language Modeling* (MLM): durante el entrenamiento, se ocultan algunas palabras aleatoriamente en el texto y BERT aprende a predecirlas usando el contexto.
- *Next Sentence Prediction* (NSP): se le muestran pares de frases y el modelo aprende a predecir si la segunda frase sigue lógicamente a la primera.

Gracias a este enfoque, BERT adquiere un conocimiento lingüístico general que luego puede adaptarse fácilmente a tareas concretas mediante una fase de ajuste o *fine-tuning*, que requiere muchos menos datos [51].

El uso de modelos como BERT presenta importantes ventajas, entre ellas destaca su capacidad para comprender el contexto completo de un texto, su versatilidad para múltiples tareas y la posibilidad de aprovechar el conocimiento ya adquirido durante el preentrenamiento. Sin embargo, también tiene algunas limitaciones, dado que su coste computacional es elevado, estando limitado a entradas de hasta 512 tokens, y su estructura interna, como ocurre en otros modelos basados en transformers, dificulta la interpretabilidad.

Además, dado que ha sido entrenado sobre lenguaje general, no está optimizado para contextos especializados como el financiero, donde el lenguaje técnico y los matices del dominio pueden afectar a la precisión del análisis. Por ello se ha optado por evaluar el rendimiento de FinBERT y FinancialBERT, variantes de BERT adaptadas al dominio financiero. Estos modelos han sido entrenados específicamente sobre textos financieros, como informes de empresas, noticias económicas o documentos regulatorios, lo que permite interpretar mejor el lenguaje y el tono característico de este ámbito. Su uso permite mantener las ventajas del modelo original, pero con una mayor sensibilidad al sector, lo que lo convierte en una herramienta especialmente adecuada para tareas como el análisis de sentimientos en titulares bursátiles.

FINBERT CLÁSICO

FinBERT es una variante del modelo BERT adaptada específicamente al dominio financiero. Fue desarrollada por Prosus AI y entrenada sobre un gran corpus de textos financieros. A diferencia del BERT general, FinBERT está diseñado para captar con mayor precisión el tono emocional de los textos financieros, lo que lo convierte en una herramienta especialmente útil para tareas como el análisis de sentimientos en titulares bursátiles [\[52\]](#) [\[53\]](#).

Inicialmente, se ha empleado FinBERT como modelo de clasificación sin aplicar ningún ajuste adicional (*fine-tuning*), aprovechando su entrenamiento previo. La implementación realizada carga tanto el modelo como el tokenizer desde el repositorio público ProsusAI/finbert y los pone en modo evaluación. A partir de ahí, se define una función que

recibe un texto, lo tokeniza y lo transforma en tensores de entrada que se pasan al modelo. La salida del modelo son los *logits*, que representan puntuaciones sin normalizar para cada clase. Estos se transforman en probabilidades mediante una función *softmax*, lo que permite interpretar directamente el grado de confianza del modelo en cada etiqueta (*positive*, *negative*, *neutral*).

Para cada titular, la función devuelve las tres probabilidades y la clase con mayor valor como predicción final. Esta lógica se aplica sobre todos los titulares del conjunto de evaluación utilizando `pandas.apply`, generando así nuevas columnas con las probabilidades y la etiqueta predicha. Además, se define un FinBERT score como la diferencia entre la probabilidad de sentimiento positivo y negativo, útil para representar la intensidad emocional del texto.

	title	sentiment	finbert_pos	finbert_neg	finbert_neu	finbert_sentiment	finbert_score
1	The international electronic industry company ...	negative	0.007849	0.972056	0.020095	negative	-0.964207
414	A tinyurl link takes users to a scamming site ...	negative	0.083786	0.022797	0.893416	neutral	0.060989
420	Compared with the FTSE 100 index , which rose ...	negative	0.146097	0.742069	0.111834	negative	-0.595972
422	Compared with the FTSE 100 index , which rose ...	negative	0.142924	0.746751	0.110326	negative	-0.603827
499	One of the challenges in the oil production in...	negative	0.014291	0.874994	0.110715	negative	-0.860703
538	Jan. 6 -- Ford is struggling in the face of sl...	negative	0.014096	0.968470	0.017435	negative	-0.954374
541	Peer Peugeot fell 0.81 pct as its sales rose o...	negative	0.008439	0.974991	0.016570	negative	-0.966552
542	Pharmaceuticals group Orion Corp reported a fa...	negative	0.007764	0.975307	0.016929	negative	-0.967543
543	However , the growth margin slowed down due to...	negative	0.015090	0.961052	0.023858	negative	-0.945962
668	2009 3 February 2010 - Finland-based steel mak...	negative	0.009730	0.975287	0.014983	negative	-0.965557
669	(ADPnews) - Feb 3 , 2010 - Finland-based ste...	negative	0.029220	0.962347	0.008433	negative	-0.933127
670	Result before taxes decreased to nearly EUR 14...	negative	0.013126	0.971760	0.015113	negative	-0.958634
671	The company slipped to an operating loss of EU...	negative	0.007680	0.974256	0.018065	negative	-0.966576
672	23 April 2010 - Finnish construction and engin...	negative	0.012425	0.976439	0.011137	negative	-0.964014
673	Cramo slipped to a pretax loss of EUR 6.7 mill...	negative	0.010131	0.973369	0.016500	negative	-0.963237

Ilustración 5: Resultados del modelo FinBERT

El rendimiento del modelo obteniendo presenta unos resultados destacables en la tarea de análisis de sentimientos. Sin necesidad de realizar *fine-tuning*, el modelo alcanza una *accuracy* global del 92%, con valores muy equilibrados en *precision*, *recall* y *F1-score* para las tres clases.

	precision	recall	f1-score	support
Negative	0.95	0.97	0.96	604
Neutral	0.93	0.86	0.89	604
Positive	0.89	0.93	0.91	604

Table 17: Resultados de clasificación de FinBERT manual

A nivel de clase, el modelo muestra un comportamiento notable en la clasificación de sentimientos negativos, alcanzando un *F1-score* de 0.96 con un *recall* del 97%, lo que indica una gran capacidad para identificar titulares negativos con alta precisión. En la clase positiva, el rendimiento también es excelente, con un *F1-score* de 0.91 y un equilibrio notable entre *precisión* (0.89) y *recall* (0.93). La clase neutral, hasta el momento más difícil de discriminar, alcanza un *F1-score* de 0.89, ligeramente inferior al resto, aunque aún muy elevado y destacable.

Estos resultados confirman la idoneidad de utilizar un modelo previamente adaptado al dominio financiero como FinBERT. Su entrenamiento especializado le permite reconocer con eficacia las características propias del lenguaje económico, incluso sin necesidad de ser reentrenado sobre el conjunto específico de titulares utilizados en este trabajo. Comparado con los enfoques ya probados, FinBERT ofrece el rendimiento más alto y equilibrado.

FINBERT JUNTO A PIPELINE DE HUGGINGFACE

Como segundo enfoque, se optó por utilizar la funcionalidad `pipeline` de la biblioteca HuggingFace Transformers, que permite aplicar modelos preentrenados de forma directa y simplificada. En este caso, se utilizó el mismo modelo FinBERT (ProsusAI/finbert) sin realizar *fine-tuning*, pero a través del pipeline de análisis de sentimientos integrado. Esta herramienta abstrae el proceso de tokenización, inferencia y decodificación, devolviendo directamente la etiqueta predicha para cada entrada de texto.

La principal ventaja de este enfoque es su facilidad de uso y su integración rápida en flujos de trabajo, permitiendo clasificar grandes volúmenes de texto con pocas líneas de código.

No obstante, al tratarse de una interfaz de alto nivel, se pierde parte del control sobre el procesamiento interno del modelo, como la extracción de probabilidades. Aun así, este enfoque es especialmente útil para verificar la robustez del modelo.

Los resultados obtenidos con la implementación mediante pipeline de HuggingFace muestran un rendimiento perfecto, con una *accuracy* del 100% y métricas de *precision*, *recall* y *F1-score* iguales a 1.00 en todas las clases.

	Precision	Recall	F1-score	Support
Negative	1.00	1.00	1.00	604
Neutral	1.00	1.00	1.00	604
Positive	1.00	1.00	1.00	604

Table 18: Resultados de clasificación de FinBERT con Pipeline

Aunque inicialmente este resultado podría interpretarse como un indicador de un ajuste perfecto del modelo, se ha querido estudiar detenidamente. El conjunto de evaluación utilizado, *Financial PhraseBank*, es un corpus muy conocido y ampliamente empleado en tareas de análisis de sentimientos financieros, tanto en publicaciones académicas como en *benchmarks*. Aunque el repositorio oficial de FinBERT (ProsusAI/finbert) no detalla explícitamente qué datos fueron utilizados durante su entrenamiento, es razonable suponer que este corpus pudo haber formado parte del conjunto de entrenamiento o evaluación interna del modelo. Esto explicaría los resultados observados, no como una señal de generalización real, sino como una coincidencia entre los datos de evaluación y el conocimiento previamente adquirido por el modelo. Es importante destacar que, dado que no se ha realizado *fine-tuning* sobre este conjunto, no puede hablarse de *overfitting*.

Estos resultados no se van a interpretar como una validación externa del modelo, sino como una verificación de su capacidad para reconocer patrones que ya ha visto anteriormente. Esta limitación se tendrá en cuenta en la comparación final entre modelos y en la construcción del ensemble propuesto.

Cabe destacar que, aunque en ambos enfoques (manual vs pipeline de HuggingFace) se implementa el mismo modelo (ProsusAI/finbert), la diferencia de rendimiento puede deberse a la forma en que se gestionan internamente las predicciones. En la implementación manual, el proceso de cálculo de probabilidades y asignación de etiquetas implica un mapeo manual de índices a clases que, si no está perfectamente alineado con la configuración del modelo, puede dar lugar a errores sutiles en la clasificación. En cambio, el pipeline automatiza este proceso siguiendo la configuración original del modelo, evitando errores de interpretación y reproduciendo de forma fiel su comportamiento previsto.

FIN-TUNING DE FINBERT

Como tercer enfoque, se optó por aplicar una técnica de *fine-tuning* sobre el modelo FinBERT. A diferencia de las aproximaciones anteriores, donde se utilizaba el modelo tal como fue publicado (es decir, únicamente en modo inferencia), el *fine-tuning* consiste en ajustar los pesos del modelo preentrenado utilizando un conjunto de datos etiquetado específico [67]. En este caso, el objetivo era especializar aún más FinBERT en la clasificación de sentimientos, logrando que aprenda los patrones propios del dataset de evaluación. Esta técnica permite aprovechar todo el conocimiento lingüístico adquirido durante el preentrenamiento, pero adaptándolo a la tarea específica que se está trabajando, con un coste computacional mucho menor que entrenar un modelo desde cero [68].

Para ello, se utilizó la clase `Trainer` de la biblioteca HuggingFace Transformers, que permite gestionar automáticamente el proceso de entrenamiento, validación y evaluación. Se dividió el conjunto de datos en entrenamiento y validación (80/20), se tokenizó con el tokenizer oficial de FinBERT, y se definió un modelo de clasificación con tres etiquetas. Se definieron 3 epoch, donde el modelo fue ajustado con un tamaño de *batch* moderado y una función de pérdida basada en entropía cruzada. Al finalizar el entrenamiento, se evaluó el modelo sobre el conjunto de validación.

Los resultados obtenidos muestran una mejora significativa respecto a los enfoques anteriores. El modelo alcanzó una *accuracy* global del 91%, así como un *macro F1-score* de 0.91, indicando un rendimiento equilibrado en las tres clases (negativa, neutral y positiva).

	Precision	Recall	F1-score	Support
Negative	0.93	0.95	0.94	128
Neutral	0.88	0.87	0.87	107
Positive	0.93	0.91	0.92	128

Table 19: Resultados de clasificación de FinBERT con *fine-tuning*

En particular, el modelo destaca por su capacidad para clasificar con alta precisión tanto sentimientos positivos como negativos, con *F1-scores* de 0.92 y 0.94, respectivamente. La clase neutral, que ya se ha observado que resulta más difícil de capturar, obtiene también un resultado razonable con un *F1-score* de 0.87.

Estos resultados reflejan que el *fine-tuning* ha permitido que el modelo se adapte con mayor precisión a las particularidades lingüísticas del conjunto de datos utilizado, mejorando la sensibilidad a los matices contextuales que definen cada clase de sentimiento. Además, el modelo demuestra ser robusto y balanceado, sin mostrar sesgos evidentes hacia una categoría específica como ocurría con otros enfoques previos.

FINANCIALBERT

Como cuarto enfoque se evaluó el modelo FinancialBERT, concretamente la versión Sigma/financial-sentiment-analysis, publicada en HuggingFace [54]. Este modelo, al igual que FinBERT, está basado en la arquitectura BERT, habiendo sido entrenado desde cero sobre grandes volúmenes de texto financiero, incluyendo informes anuales, noticias económicas y análisis bursátiles. Su objetivo es capturar las expresiones características del lenguaje financiero sin requerir un proceso adicional de *fine-tuning* para tareas específicas.

A diferencia de FinBERT, que fue preentrenado con textos financieros pero pensado para adaptarse mediante *fine-tuning*, FinancialBERT ya viene afinado para clasificación de

sentimiento. Por ello, se va a evaluar el modelo directamente para inferencia, sin realizar ajustes adicionales, estudiando su rendimiento tal y como se ha publicado.

La implementación consistió en tokenizar los titulares mediante el BertTokenizer correspondiente, procesarlos en lotes con un DataLoader y obtener las predicciones directamente del modelo en modo evaluación (`eval()`). Posteriormente, las etiquetas predichas fueron comparadas con las etiquetas reales del conjunto de evaluación para analizar el comportamiento del modelo.

Los resultados obtenidos muestran un rendimiento sólido pero inferior al modelo FinBERT fine-tuneado. Se alcanzó una *accuracy* del 82% y un *macro F1-score* de 0.82, lo cual confirma que el modelo generaliza razonablemente bien.

	Precision	Recall	F1-score	Support
Negative	0.95	0.77	0.85	604
Neutral	0.69	0.90	0.78	604
Positive	0.87	0.79	0.83	604

Table 20: Resultados de clasificación de FinacialBERT

Se observa un comportamiento más desigual entre clases en comparación con otros modelos evaluados. En concreto, en la clase neutral es donde se observa el mayor desequilibrio. Aunque obtiene un *recall* muy alto (0.90), su precisión es significativamente más baja (0.69), lo que sugiere que el modelo tiende a clasificar en exceso como "neutral" incluso cuando no corresponde. El *F1-score* (0.78) refleja la dificultad que presenta el modelo para identificar correctamente los límites semánticos de esta clase intermedia.

En cuanto a las clases positiva y negativa, el modelo muestra un equilibrio relativamente mejor que con la neutral. Aunque algo más conservador que en la clase negativa, se muestra razonablemente eficaz al identificar titulares con sentimiento positivo, aunque sigue existiendo una confusión notable con la clase neutral.

Se puede concluir que FinancialBERT ofrece una alternativa rápida y sin necesidad de entrenamiento adicional para tareas de análisis de sentimiento financiero. Sin embargo, sus resultados sugieren que el ajuste fino sigue siendo una estrategia más efectiva cuando se dispone de un conjunto de datos etiquetado de calidad.

FIN-TUNING DE FINANCIABERT

Aunque el modelo FinancialBERT (ahmedrachid/FinancialBERT-Sentiment-Analysis) [55] fue diseñado específicamente para el análisis de sentimiento en textos financieros, se ha decidido realizar un proceso de *fine-tuning* adicional con el objetivo de evaluar si una adaptación al conjunto de datos utilizado podría mejorar su rendimiento. A pesar de estar previamente entrenado sobre corpus especializados, es posible que ciertas expresiones o estilos de redacción del dataset utilizado no estuvieran completamente representados en el preentrenamiento original. El *fine-tuning* permite refinar el modelo sobre la distribución exacta del problema a resolver, potenciando así su capacidad de generalización sobre ese dominio específico.

La implementación siguió una estructura similar al entrenamiento de otros modelos BERT: se reutilizó el tokenizer del modelo original, se construyeron datasets personalizados con `torch.utils.data.Dataset`, y se gestionó el entrenamiento mediante un optimizador AdamW y un scheduler lineal. Se entrenó durante tres épocas con batch size moderado, empleando validación cruzada sobre un 20% del dataset para monitorizar el rendimiento durante el ajuste. Los resultados tras el *fine-tuning* muestran una mejora significativa respecto a la versión sin ajuste:

	Precision	Recall	F1-score	Support
Negative	0.98	0.98	0.98	128
Neutral	0.91	0.97	0.94	107
Positive	0.97	0.91	0.94	128

Table 21: Resultados de clasificación de FinancialBERT con Fin-Tuning

Se obtuvo una *accuracy* del 96% y un *macro F1-score* también del 96%, lo que representa el mejor rendimiento global entre todos los enfoques evaluados. A nivel de clases, el modelo alcanzó un *F1-score* de 0.98 en la clase negativa, con *precision* y *recall* igualmente elevados (0.98), lo que indica una detección casi perfecta de titulares con sentimiento claramente negativo. Para la clase neutral, que es la que mayor dificultad ha supuesto por su ambigüedad semántica, se ha logrado un *F1-score* de 0.94, con *precision* del 0.91 y un *recall* del 0.97. Esto sugiere que el *fine-tuning* ha permitido al modelo capturar mejor a esta categoría, evitando tanto falsos positivos.

Los resultados obtenidos muestran, no solo que el modelo es competente de base, sino que el proceso de *fine-tuning* ha sido efectivo para ajustarlo aún más a la distribución específica del conjunto de datos de evaluación, adaptándose a patrones lingüísticos concretos presentes en los titulares utilizados. Así, el modelo ha logrado corregir gran parte del sesgo hacia la clase neutral que afectaba al modelo sin ajuste, y ha mejorado la sensibilidad en las tres clases sin sacrificar precisión. Este caso demuestra que, incluso en modelos ya especializados como FinancialBERT, realizar un ajuste fino sobre datos reales de aplicación puede traducirse en mejoras sustanciales, especialmente cuando se busca robustez en contextos reales de análisis financiero.

LLM

Los *Large Language Models* (LLM) son modelos de lenguaje de gran escala entrenados con cantidades masivas de texto y parámetros, capaces de generar, completar, traducir y analizar texto con un alto grado de comprensión semántica. A diferencia de modelos tradicionales o redes neuronales recurrentes, los LLM están basados en arquitecturas Transformer de gran capacidad y profundidad, como FinGPT o LLaMA, lo que les permite identificar relaciones complejas en el lenguaje y realizar tareas avanzadas sin necesidad de entrenamiento específico para cada una de ellas [56].

Como último paso, se exploró la posibilidad de estudiar el comportamiento de distintos LLM para el análisis de sentimiento financiero. Se barajaron varias alternativas, entre ellas **LLaMA** (Meta), **LoRA** (Low-Rank Adaptation) o **FinGPT** [57]. La combinación de LLaMA + LoRA se ha comprobado que resulta eficaz para tareas específicas, incluyendo análisis de sentimiento, sin necesidad de reentrenar completamente el modelo base. FinGPT es una variante especializada de LLM desarrollada específicamente para tareas financieras

A nivel técnico, se diseñó un entorno en Python para cargar modelos como `internlm-20b` y variantes de FinGPT con soporte LoRA, empleando técnicas de optimización como `device_map="auto"` y `offload_folder` para gestionar la carga en CPU con recursos limitados. Sin embargo, las limitaciones de memoria disponibles impidieron completar la carga y ejecución de los distintos modelos en el entorno de trabajo, incluso con optimizaciones como el vaciado de caché o la ejecución en modo evaluación (`model.eval()`).

Por estos motivos, no se pudo integrar ningún LLM como parte funcional del sistema final. El uso de FinGPT o modelos de tipo LLaMA adaptados al dominio financiero podría mejorar significativamente la capacidad de generalización del sistema, especialmente en escenarios reales donde el lenguaje de las noticias es más complejo o ambiguo, por lo que se planteará su implementación como trabajo futuro.

ENSEMBLE

La técnica conocida como *ensemble*, o combinación de modelos, es ampliamente utilizada en aprendizaje automático para mejorar tanto el rendimiento como la robustez de los sistemas predictivos. Esta estrategia consiste en agrupar las predicciones de varios modelos individuales, de forma que el resultado final se obtenga mediante un mecanismo de agregación (como votación mayoritaria o ponderada). La lógica detrás de esta técnica es que diferentes modelos pueden cometer errores distintos en distintas instancias del conjunto de datos, por lo que combinar sus resultados permite reducir el impacto de errores individuales, mejorar la estabilidad de la predicción y, en muchos casos, aumentar el rendimiento general del sistema [58] [59]. En el contexto del análisis de sentimiento, donde las clases pueden solaparse o ser ambiguas (como ocurre con el sentimiento neutral), un ensemble permite capturar perspectivas complementarias de modelos entrenados con arquitecturas o datos ligeramente distintos.

Para la combinación final de modelos se optó por utilizar un enfoque de votación mayoritaria (*hard voting*). Este método consiste en tomar la predicción más frecuente entre varios modelos para cada instancia del conjunto de datos. La elección de este tipo de ensemble se basa en varios factores. Por un lado, todos los modelos considerados producen salidas categóricas (etiquetas de sentimiento), por lo que la votación resulta aplicable. Además, al contar con tres modelos, siempre existe una mayoría clara sin necesidad de desempates adicionales. A diferencia de otros métodos como la votación ponderada o el stacking, que requieren una calibración adicional o un modelo meta, la votación mayoritaria es una técnica robusta, sencilla de implementar e interpretable, lo que la convierte en una opción ideal en contextos académicos. Este ensemble permite combinar las fortalezas de cada modelo, disminuyendo sus errores individuales y mejorando la estabilidad de las predicciones finales.

Tras analizar el rendimiento de los distintos modelos aplicados al análisis de sentimiento financiero, se ha decidido incluir en el ensemble final los siguientes tres: **FinBERT fine-tuneado**, **FinancialBERT fine-tuneado** y **FinancialBERT preentrenado** (Sigma). La

decisión se basa no solo en los resultados cuantitativos obtenidos, sino también en criterios de fiabilidad, generalización y complementariedad entre modelos. Aunque modelos como FinBERT sin ajuste (vía pipeline) ofrecieron métricas aparentemente perfectas, fueron descartados debido a la alta posibilidad de que su entrenamiento incluyera el mismo conjunto de datos utilizado en la evaluación (*FinancialPhraseBank*), lo que comprometería la validez de los resultados por riesgo de *data leakage*, haciendo que el modelo memorice patrones en lugar de generalizar. No obstante, dado que se ha realizado un proceso controlado de *fine-tuning* sobre FinBERT utilizando exclusivamente el conjunto etiquetado seleccionado, se considera legítimo incluir esta versión ajustada, ya que su rendimiento refleja una adaptación específica al problema y no únicamente una inferencia directa sobre datos potencialmente vistos.

Por otra parte, FinancialBERT fine-tuneado ofrece resultados sólidos, equilibrados y sin indicios de sobreajuste. Además, incluir FinancialBERT preentrenado (Sigma), aunque con métricas más modestas, añade diversidad al ensemble, aportando otra perspectiva semántica que contribuye a reducir errores sistemáticos. Así, la combinación de estos tres modelos ofrece una solución robusta, bien fundamentada y con alto potencial de generalización en entornos reales de análisis de noticias financieras.

El ensemble final, construido mediante votación mayoritaria entre los modelos FinBERT fine-tuneado, FinancialBERT fine-tuneado y FinancialBERT preentrenado (Sigma), ha alcanzado un rendimiento global excelente, con una *accuracy* del 98% y un *macro F1-score* de 0.98:

	Precision	Recall	F1-score	Support
Negative	0.98	0.99	0.99	604
Neutral	0.97	0.98	0.97	604
Positive	0.99	0.97	0.98	604

Table 22: Resultados de clasificación de Ensemble

Estos resultados no solo superan los obtenidos por los modelos individuales de forma aislada, sino que además reflejan una consistencia entre clases, con *F1-scores* de 0.99 (negativa), 0.97 (neutral) y 0.98 (positiva). La matriz de confusión confirma esta homogeneidad, mostrando una distribución de errores mínima y equilibrada:

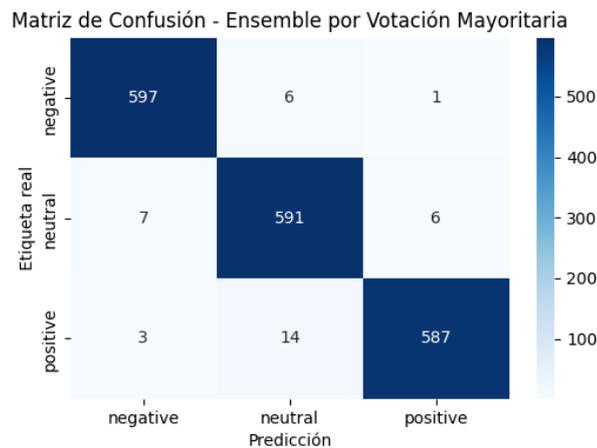


Ilustración 6: Matriz de confusión del modelo ensemble basado en votación mayoritaria

Este rendimiento puede atribuirse a la capacidad del ensemble para minimizar los sesgos individuales de cada modelo, aprovechando la especialización de cada uno y corrigiendo errores sistemáticos. Por ejemplo, errores cometidos por el modelo Sigma en la clase positiva o las ligeras desviaciones de FinBERT en la clase neutral se ven compensados por el consenso del resto. De este modo, el sistema combinado logra un equilibrio difícil de alcanzar por modelos individuales, reafirmando el valor de los enfoques colaborativos cuando se trabaja con problemas de clasificación complejos y de alto impacto como el análisis de sentimiento financiero.

El **modelo** seleccionado para ser utilizado en la fase **final** de generación de informes es el **ensemble**. Esta decisión se justifica tanto por su rendimiento cuantitativo, con una *accuracy* global del 98% y un *macro F1-score* de 0.98, como por su solidez frente a sesgos individuales y su capacidad para mantener una alta precisión de forma homogénea en las tres clases analizadas.

Capítulo 5. GENERACIÓN DE INFORMES

Una vez finalizado el desarrollo del sistema de análisis de sentimientos sobre titulares financieros, el siguiente paso fue crear una aplicación que permitiera visualizar de forma clara la información generada para una empresa concreta. Inicialmente se implementó Dash por su capacidad para construir interfaces web personalizables. No obstante, para la versión final se migró a Streamlit, ya que su desarrollo es más ágil, su visualización más clara, y se adapta mejor a aplicaciones orientadas a informes interactivos.

La aplicación no se limita a mostrar el resultado del análisis de una noticia concreta, sino que permite seleccionar una empresa y un periodo de tiempo específicos para analizar en profundidad qué tono predomina en ese intervalo, cuáles son las palabras más repetidas, cómo evoluciona el sentimiento del mercado, y qué temas se están tratando según cada tipo de sentimiento. Además, se incluye un resumen automático de las noticias de ese periodo. El objetivo que se persigue es diseñar un *dashboard* completo que permita explorar de forma sencilla pero detallada el comportamiento del mercado en torno a una empresa concreta, facilitando el análisis y la toma de decisiones.

Esta herramienta tiene como objetivo transformar una gran cantidad de titulares financieros, los cuales sería muy costosos de analizar manualmente, en información estructurada y visualmente clara, enfocado en torno a una empresa concreta durante un periodo de tiempo determinado. Cabe destacar que, aunque el análisis de sentimientos es una parte importante, la aplicación integra distintas capas de análisis que, en conjunto, permiten entender mejor el contexto de una compañía y detectar señales que podrían tener un impacto en la toma de decisiones. El objetivo es ofrecer una visión global del entorno informativo que rodea a una empresa.

La aplicación desarrollada se estructura en torno a una pantalla principal o de inicio, que permite al usuario acceder libremente a cada uno de los módulos de análisis disponibles a través del menú lateral. A diferencia de un flujo secuencial clásico, cada módulo ha sido diseñado para funcionar de manera independiente, permitiendo así una exploración flexible y adaptada a las necesidades del usuario. La navegación parte de un menú central que da acceso a los siguientes cuatro módulos principales:

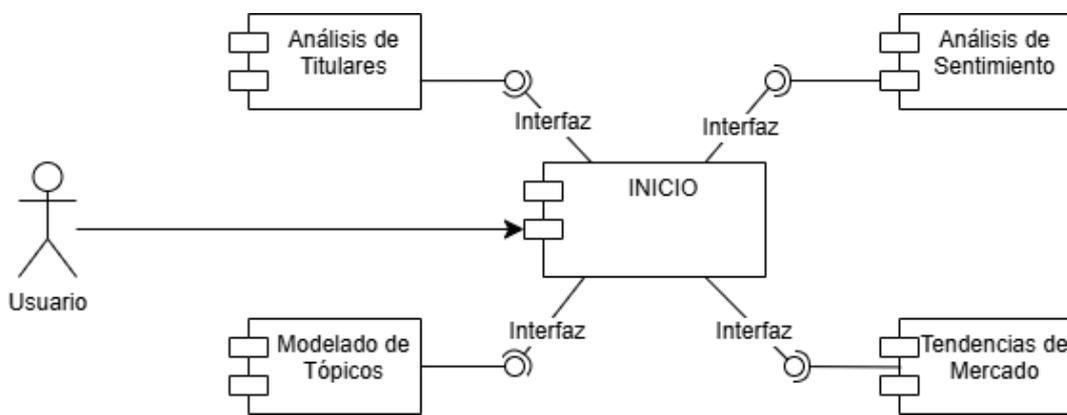


Ilustración 7: Diagrama de componentes de la aplicación

1. **Análisis de Titulares:** En este módulo se introduce al usuario en el conjunto de titulares y facilita una primera exploración visual del contenido textual.
2. **Análisis de Sentimiento:** Se enfoca en el análisis emocional de los textos, mostrando la distribución y evolución del sentimiento de los titulares seleccionados.
3. **Modelado de Tópicos:** Se aplican técnicas de agrupamiento temático (*topic modeling*) para identificar los principales tópicos tratados en los titulares, diferenciando según sentimiento.
4. **Tendencias de Mercado:** El cuarto módulo cruza la información semántica con datos financieros estructurados, ofreciendo una visión conjunta del comportamiento del mercado y el tono informativo asociado.

A continuación, se expone cómo se ha desarrollado cada componente e integrado en la interfaz de usuario.

SISTEMA DE FILTRADO

Permitir al usuario aislar tanto la entidad como el intervalo temporal aporta personalización y refuerza la capacidad de la herramienta para extraer conclusiones relevantes. Este módulo permite al usuario seleccionar de forma interactiva una compañía concreta (a través de su *ticker*) y centrar el análisis a un periodo temporal específico.

Estos filtros mejoran la usabilidad general de la aplicación y simplifican la exploración de datos. Ambos se aplican directamente sobre el dataset de titulares financieros y condicionan todo el análisis posterior, desde las palabras clave hasta los modelos de sentimiento, ya que solo se mostrará la información relativa a dicha selección.

Para la selección del *ticker* se ha utilizado el desplegable (*selectbox*) de Streamlit, alimentado con los valores únicos de la columna *stock*. Para la fecha, se ha empleado un *date_input* con opción de rango doble. En lugar de técnicas más complejas de búsqueda o filtrado dinámico, se ha optado por un enfoque directo que ofrece una experiencia ágil y robusta.

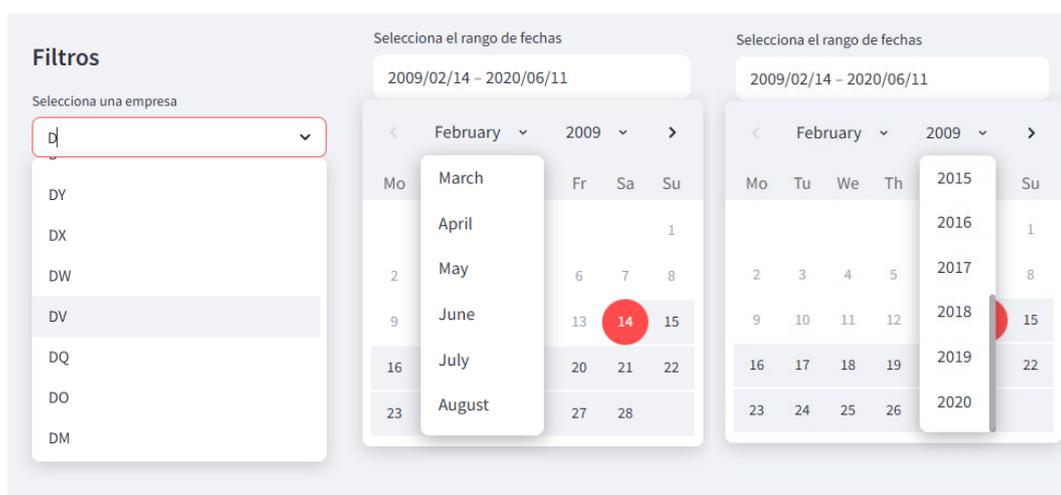


Ilustración 8: Ejemplo sistema de filtrado

KEYWORDS

Este módulo muestra las palabras que más se repiten en los titulares financieros de una empresa en un periodo concreto. Permite ver rápidamente de qué se está hablando en el periodo seleccionado, es decir, sirve como resumen visual.

Se extraen las palabras clave más repetidas de los titulares usando un contador de frecuencia (Counter) tras aplicar una limpieza básica a los datos. Primero se normalizan los titulares convirtiéndolos a minúsculas, eliminando URLs y caracteres especiales, y realizando una tokenización básica. A continuación, se filtran las palabras vacías (*stopwords*) y se aplica lematización para agrupar variantes de una misma raíz (por ejemplo, *announces* y *announced* se reducen a *announce*). También se eliminan tokens considerados ruido, como números aislados o letras sueltas. Integrándolo todo en la interfaz se obtiene:

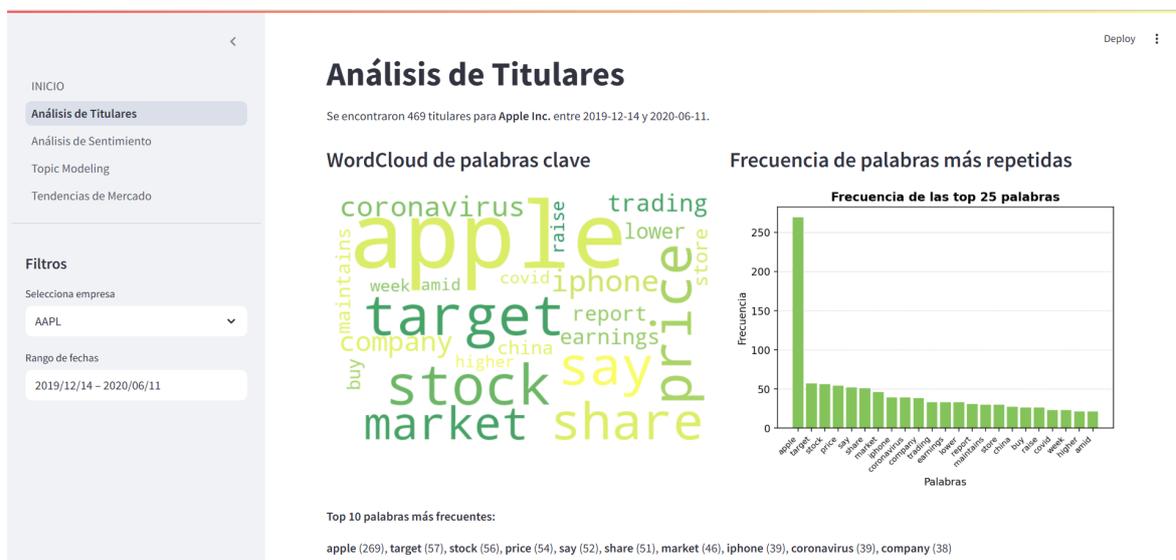


Ilustración 9: Interfaz de Análisis de Titulares

Todo esto se actualiza dinámicamente con Streamlit según los filtros de empresa y periodo seleccionados. Se ha optado por esta solución porque es rápida, clara y visualmente efectiva, sin requerir procesamiento pesado.

SENTIMENT ANALYSIS

El análisis de sentimiento se ha incorporado como una pieza clave de la interfaz, ya que permite evaluar e interpretar el tono emocional de los titulares financieros de forma rápida y directa, sin necesidad de evaluar cada titular individualmente. Dado que las decisiones del mercado no solo se rigen por datos cuantitativos sino también por percepciones, expectativas y emociones, poder medir automáticamente el tono (positivo, negativo o neutro) de las noticias aporta una capa adicional de comprensión y contexto. Esta información es especialmente útil tanto para analistas como para usuarios no expertos, ya que permite observar tendencias cualitativas que pueden anticipar movimientos del mercado o cambios de percepción sobre una empresa. Además, la agregación temporal del sentimiento permite observar cómo evoluciona temporalmente, lo que resulta clave para detectar eventos anómalos o patrones.

Se han incorporado varias visualizaciones interactivas que permiten al usuario explorar tanto la distribución general como la evolución temporal del tono de los titulares. En primer lugar, se ha incluido un gráfico de barras y otro de pastel que muestran el reparto de titulares en cada clase (positivo, neutral y negativo) dentro del periodo seleccionado, lo que permite detectar rápidamente si predomina una percepción favorable o desfavorable de la empresa.

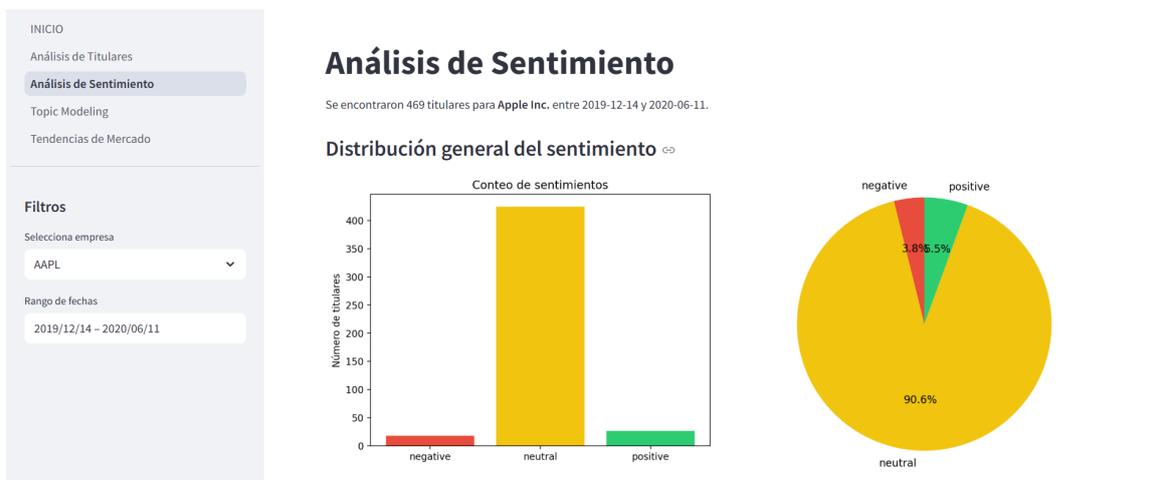


Ilustración 10: Interfaz Análisis de Sentimiento

A través de un gráfico de área apilada se ha representado cómo varía día a día el número de titulares clasificados por sentimiento. Cada color corresponde a una categoría (positivo, neutral o negativo), lo que facilita detectar visualmente tendencias o momentos concretos.

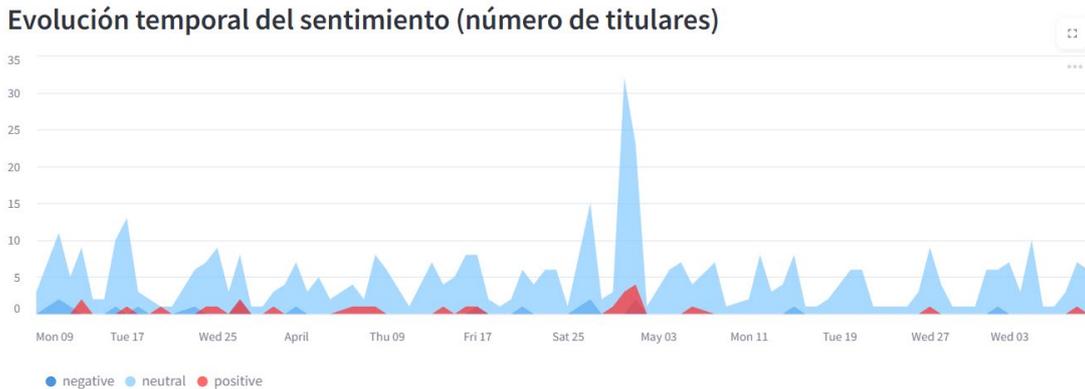


Ilustración 11: Evolución temporal del número de titulares clasificados por sentimiento

Para ofrecer una medida sintética del tono emocional de cada día, se ha definido un índice de sentimiento medio, asignando valores numéricos a cada clase (*positivo* = +1, *neutral* = 0, *negativo* = -1). La media de estos valores por día se representa mediante una línea temporal, que permite observar claramente los días donde se alcanzaron sus picos más altos. Esta gráfica permite detectar tanto cambios progresivos como anomalías puntuales, siendo útil para anticipar reacciones del mercado o analizar correlaciones con eventos externos.



Ilustración 12: Evolución del sentimiento medio diario a partir del análisis de titulares financieros.

Finalmente, se ha incluido una sección donde se identifican los cinco días donde el sentimiento medio fue más extremo, tanto en el sentido positivo como negativo. El cálculo se basa en el mismo índice definido anteriormente (positivo = +1, neutral = 0, negativo = -1), y permite destacar aquellos momentos en los que el tono resaltó especialmente. La información se presenta en formato de tabla para facilitar una lectura rápida y ordenada, con las fechas y los valores promedio correspondientes.

Top 5 días con sentimiento más positivo y más negativo

Días más positivos

date	Sentimiento medio
2020-03-20	0.5
2020-04-07	0.333
2020-04-29	0.25
2020-03-30	0.25
2020-04-06	0.2

Días más negativos

date	Sentimiento medio
2020-03-18	-0.25
2020-03-12	-0.167
2020-03-11	-0.154
2020-06-02	-0.143
2020-04-21	-0.143

Ilustración 13: Top 5 días con sentimiento medio más positivo y más negativo según los titulares financieros.

Esta visualización es especialmente útil para contextualizar picos observados en las gráficas anteriores, y para identificar momentos de especial relevancia informativa que pueden haber tenido impacto en la percepción pública o en el comportamiento del mercado.

TOPIC MODELING

La técnica de *Topic modeling* permite descubrir automáticamente los temas más frecuentes que aparecen en los titulares, sin necesidad de una clasificación previa o un listado de palabras clave. Este módulo se ha incluido porque aporta una capa de análisis más profunda, especialmente útil para entender qué se está diciendo realmente dentro de cada grupo de sentimiento. Mientras que el análisis de sentimiento muestra el tono de las noticias, el modelado de tópicos revela su contenido. Esto es especialmente útil cuando el número de titulares es alto o cuando se quiere ir más allá de las palabras sueltas del wordcloud. Además,

segmentar los temas por sentimiento permite ver con claridad si, por ejemplo, las noticias negativas giran en torno a pérdidas financieras, conflictos legales o cambios de gobierno.

El modelado de tópicos es una técnica de aprendizaje no supervisado que agrupa textos según los patrones semánticos que comparten. En este proyecto se aplica directamente sobre los titulares, una vez preprocesados, y se realiza por separado para cada tipo de sentimiento (positivo, neutral y negativo). Para cada grupo, se generan varios temas (*topics*), que consisten en listas de palabras clave representativas. Estos temas se visualizan mediante nubes de palabras, facilitando su interpretación visual. A su vez, se va a acompañar de un resumen generado automáticamente utilizando el modelo `google/flan-t5-base`, que transforma las listas de palabras clave extraídas por BERTopic en descripciones breves y comprensibles en lenguaje natural [60]. A diferencia de modelos clásicos como LDA, BERTopic utiliza representaciones vectoriales modernas (derivadas de modelos como BERT) para capturar el significado profundo de los textos. Esto permite trabajar con textos muy cortos como los titulares y detectar patrones más coherentes. Además, permite controlar el número mínimo de documentos por tema y ofrece una API muy flexible para integración en aplicaciones. Se descartó el uso de LDA por su menor rendimiento en textos breves y por requerir un número fijo de temas.

La interfaz del módulo de modelado de tópicos permite establecer el número mínimo de documentos requerido para que un grupo temático sea considerado como tópico válido, lo cual ayuda a eliminar agrupaciones poco representativas. Asimismo, incorpora un sistema de filtrado que permite seleccionar la empresa de interés y definir un rango temporal concreto. Una vez configurados los filtros, el usuario puede ejecutar el análisis haciendo clic en el botón “Ejecutar modelado de tópicos por sentimiento”, lo que activa el algoritmo de clustering y visualización. El análisis se segmenta automáticamente según la clase de sentimiento (positivo, neutral o negativo), generando tópicos independientes para cada grupo. Para cada uno de ellos se muestra una nube de palabras representativas junto a un resumen en lenguaje natural, lo que facilita la interpretación de los temas detectados sin necesidad de intervención manual.

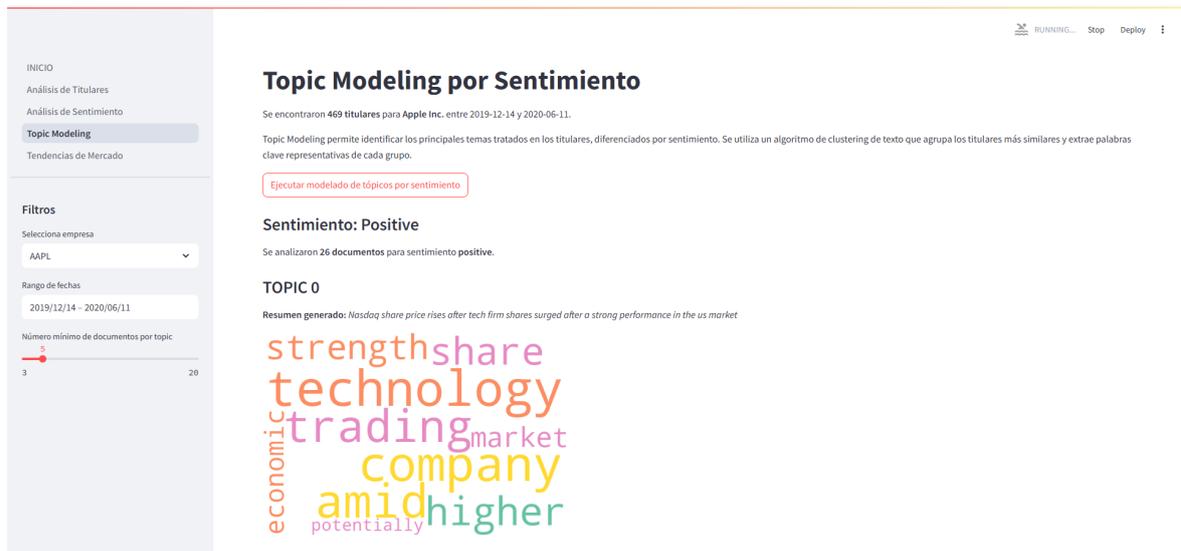


Ilustración 14: Interfaz Topic Modeling por Sentimiento

EVOLUCIÓN TEMPORAL DEL MERCADO

Se ha decidido incluir un módulo específico que relacione el sentimiento expresado en los titulares con la evolución real de las cotizaciones del mercado. Lo que se pretende conseguir es ofrecer una visión conjunta entre los datos no estructurados (textos) y los datos estructurados (precios y volúmenes), permitiendo analizar si existe una correlación visible entre ambos. Esta funcionalidad es especialmente útil para analistas financieros, ya que facilita la interpretación del impacto potencial del tono informativo en el comportamiento del mercado. Además, proporciona un valor añadido respecto a los módulos anteriores, que se centran únicamente en el análisis textual.

Para representar la evolución temporal, se emplea un enfoque simple pero efectivo basado en la agregación las predicciones de sentimiento obtenidas previamente mediante el ensemble de modelos BERT. En paralelo, se recuperan los datos históricos de mercado a través de la librería `yfinance`, extrayendo los precios de cierre (`Close`) y los volúmenes negociados (`Volume`). Los datos se combinan utilizando la fecha como clave común, asegurando la alineación temporal.

Se muestra la evolución diaria del precio de cierre, lo que permite identificar tendencias y variaciones significativas a lo largo del período seleccionado. A continuación, se representa el volumen de negociación diario, útil para contextualizar el nivel de actividad en el mercado y detectar posibles picos relacionados con eventos relevantes.

Tendencias de Mercado

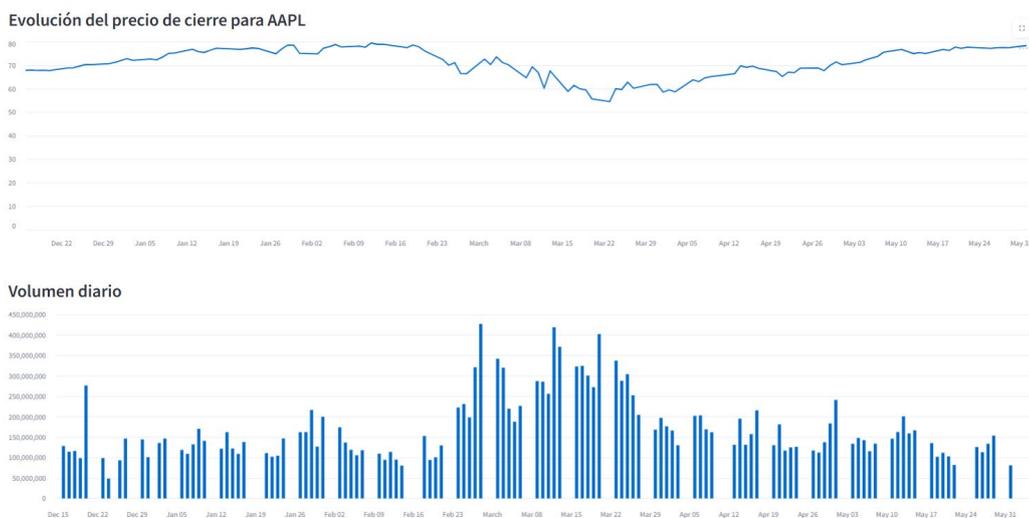


Ilustración 15: Reresentación visual de las Tecendencia de Mercado

A su vez, se presenta un gráfico comparativo que superpone el sentimiento medio diario (calculado a partir de las predicciones del ensemble) con el precio de cierre, permitiendo observar posibles correlaciones entre la percepción informativa y la evolución real del activo.

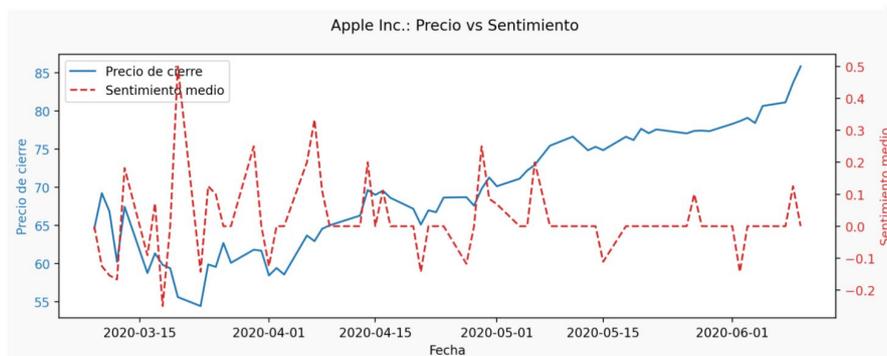


Ilustración 16: Comparación temporal Precio vs Sentimiento

RESUMEN DE LAS IDEAS PRINCIPALES

El resumen automático se ha incluido como cierre del análisis para proporcionar una comprensión rápida de los titulares del periodo seleccionado. Aunque las visualizaciones permiten explorar patrones y tendencias, este módulo añade una capa de síntesis que aporta valor especialmente en contextos donde se trabaja con un alto volumen de información o se necesita elaborar informes. Permite al usuario obtener en pocos segundos una idea general del tono y el contenido predominante en las noticias, sin necesidad de leerlas una a una. Además, mejora la experiencia final de uso, aportando una funcionalidad de lenguaje natural que complementa perfectamente los módulos anteriores.

Se ha utilizado el modelo `csebuetnlp/mT5_multilingual_XLSum` [61], un modelo preentrenado capaz de generar descripciones breves en lenguaje natural a partir de textos concatenados. Este enfoque se ha elegido por su capacidad de generar resúmenes informativos y coherentes incluso en contextos financieros, sin necesidad de entrenamiento adicional. Gracias a su eficiencia y rendimiento, permite sintetizar los titulares más relevantes de cada período y contextualizar los patrones observados en los gráficos.

Este módulo toma todos los titulares asociados a una empresa en un periodo determinado y genera un resumen corto y coherente en lenguaje natural sobre los temas de los que tratan los distintos titulares. Para ello, concatena los titulares en bruto y los envía a un modelo de generación de texto entrenado para condensar grandes volúmenes de información en pocas frases. El resultado es un texto breve que recoge las ideas principales tratadas en los titulares.

Resumen de los titulares

Apple's shares have jumped to their highest level for the first time in more than a decade. Here are some of the key moments in the stock market - and what does this mean for Apple?

Ilustración 17: Resumen automático generado a partir de los titulares financieros

INTEGRACIÓN FINAL EN LA INTERFAZ

La decisión de utilizar Streamlit se basa en su facilidad de desarrollo, su flujo más natural para prototipado rápido y su integración directa con entornos de ciencia de datos como Jupyter, Colab o scripts de Python. A diferencia de Dash, Streamlit no requiere construir *callbacks* explícitos, ni manejar estructuras complejas de *layout* ni componentes reactivos. Esto acelera significativamente el proceso de desarrollo y depuración, especialmente cuando se combinan múltiples módulos de análisis con modelos de NLP, gráficas y preprocesamientos [\[62\]](#) [\[63\]](#).

Otra ventaja clave es su sistema de múltiples páginas (*pages/*), que permite dividir la aplicación en secciones modulares, muy útiles para mantener la claridad visual y organizativa. Además, Streamlit proporciona una experiencia de usuario muy fluida, permitiendo añadir gráficos, controles y modelos sin necesidad de construir una arquitectura *front-end* completa. A su vez, tiene mejor soporte para componentes interactivos modernos (*sliders*, *selectboxes*, *inputs*), y una integración sencilla con herramientas de caché, gráficos en *matplotlib* y *plotly*, y despliegue directo en plataformas como Streamlit Cloud.

Todos los módulos de análisis están completamente integrados dentro de la aplicación y comparten una lógica común basada en filtros por empresa y rango temporal. Esto garantiza la coherencia entre los resultados mostrados en cada página y permite al usuario navegar de forma fluida. Por ejemplo, al seleccionar una empresa concreta, se puede visualizar primero su volumen informativo (*keywords*), luego la polaridad de los titulares (sentimiento), explorar los temas subyacentes por clase emocional (*topics*), y finalmente observar cómo todo esto se relaciona con la evolución temporal real del mercado. De esta manera, la interfaz actúa como una herramienta analítica cohesionada y flexible, que guía al usuario a través de un flujo natural de exploración e interpretación.

La aplicación ha sido estructurada siguiendo una lógica modular, dividiendo los distintos bloques de análisis en cuatro páginas principales, todas accesibles desde el menú lateral.

Cada una está implementada como un archivo independiente dentro de la carpeta pages/, y responde a un propósito analítico específico:

- **1_Análisis_de_Titulares.py:** Muestra el número total de titulares para la empresa seleccionada, la nube de palabras generada con los términos más frecuentes y un gráfico de frecuencia de palabras.
- **2_Análisis_de_Sentimiento.py:** Presenta las distribuciones de sentimiento (positivo, neutral, negativo) ya calculadas previamente mediante ensemble de modelos BERT, y su evolución en el tiempo. Se muestran gráficas de barras, pastel, evolución temporal y promedio diario.
- **3_Modelado_de_Tópicos.py:** Aplica BERTopic por separado a cada clase de sentimiento para extraer los principales temas tratados en los titulares. Se presentan los tópicos como listas de palabras clave y como nubes de palabras.
- **4_Tendencias_de_Mercado.py:** Integra datos históricos de cotización de las empresas con los indicadores de sentimiento para visualizar la relación entre el tono de las noticias y la evolución del precio. Incluye también un resumen automático generado a partir de los titulares.

Por último, el archivo INICIO.py actúa como página de inicio. No contiene lógica funcional, sino una explicación general de la herramienta, los módulos disponibles y su propósito.



Ilustración 18: Interfaz de Inicio

Capítulo 6. ANÁLISIS DE RESULTADOS

Este capítulo tiene como objetivo evaluar los resultados obtenidos a lo largo del desarrollo del sistema propuesto, tanto desde el punto de vista del rendimiento de los modelos de análisis de sentimientos como desde la perspectiva funcional de la interfaz de usuario.

En primer lugar, se van a comparar los diferentes enfoques implementados en el [Capítulo 4](#), desde modelos tradicionales hasta arquitecturas avanzadas basadas en transformers, con el fin de determinar cuál ofrece un mejor equilibrio entre precisión, robustez y aplicabilidad en el dominio financiero. En segundo lugar, se va a examinar el resultado final del sistema de generación de informes descrito en el [Capítulo 5](#). Se va a analizar la usabilidad, la claridad y el valor aportado por cada módulo de la interfaz.

EVALUACIÓN DE MODELOS DE CLASIFICACIÓN DE SENTIMIENTO

Los resultados obtenidos reflejan una clara diferencia de rendimiento entre los modelos clásicos basados en diccionarios o redes neuronales simples, frente a los modelos preentrenados basados en transformers. En términos de exactitud, los modelos FinBERT fine-tuneado, FinancialBERT fine-tuneado y el ensemble alcanzan valores cercanos al 97%, destacando significativamente frente al resto de enfoques.

MODELOS CLÁSICOS

Los modelos basados en léxicos como LM y VADER muestran un rendimiento cercano al azar (~50%), lo que evidencia las limitaciones de estos métodos en un contexto financiero especializado. El ajuste de puntuaciones en el modelo LM (“LM Ajustado”) mejora ligeramente el rendimiento, pero sigue sin superar a enfoques basados en aprendizaje automático.

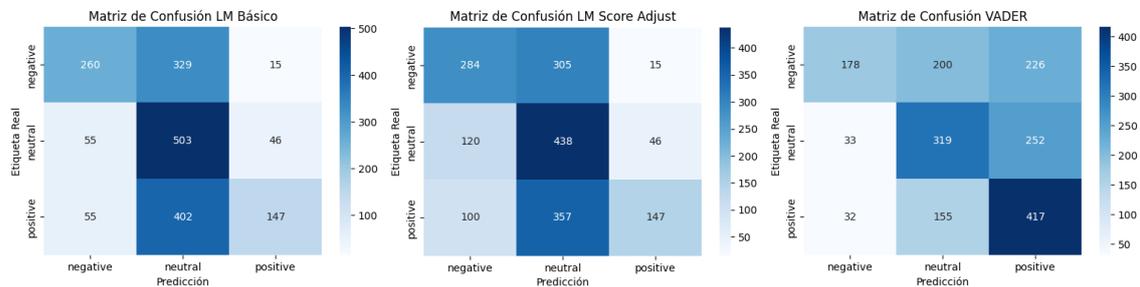


Ilustración 19: Matriz de confusión del enfoque basado en diccionarios

MODELOS LSTM

El modelo LSTM básico logra una precisión global del 52.8%, mostrando una mejora frente a los enfoques no supervisados, pero con limitaciones en la clase positiva. Sin embargo, las extensiones con embeddings preentrenados, tanto GloVe como FastText, logran una mejora sustancial en todas las métricas. El modelo LSTM + GloVe alcanza una precisión del 70%, mientras que LSTM + FastText se sitúa en el 68%.

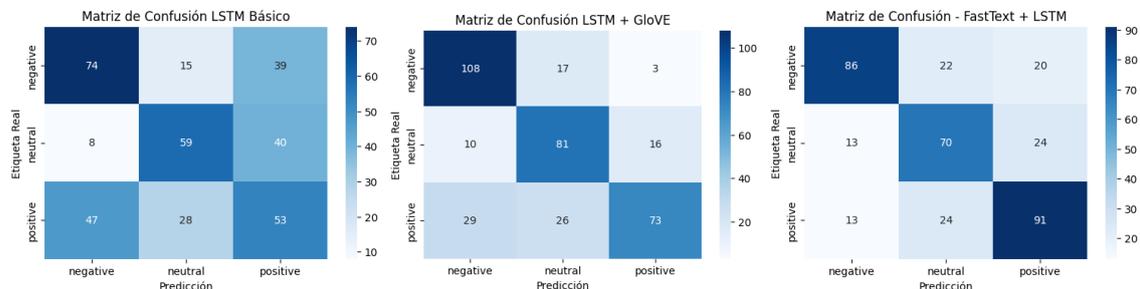


Ilustración 20: Matriz de confusión del enfoque basado en LSTM

MODELOS BASADOS EN TRANSFORMERS

FinBERT muestra una gran capacidad de generalización incluso sin *fine-tuning* (92% de *accuracy*), lo que refuerza su ajuste al dominio financiero. Tras el *fine-tuning*, se alcanza un rendimiento máximo (97.4%). Se obtienen resultados similares con FinancialBERT, lo que sugiere que ambos modelos son adecuados para este tipo de tarea. El modelo Sigma, aunque entrenado en un conjunto de datos distinto, también muestra buena *precision* (82%).

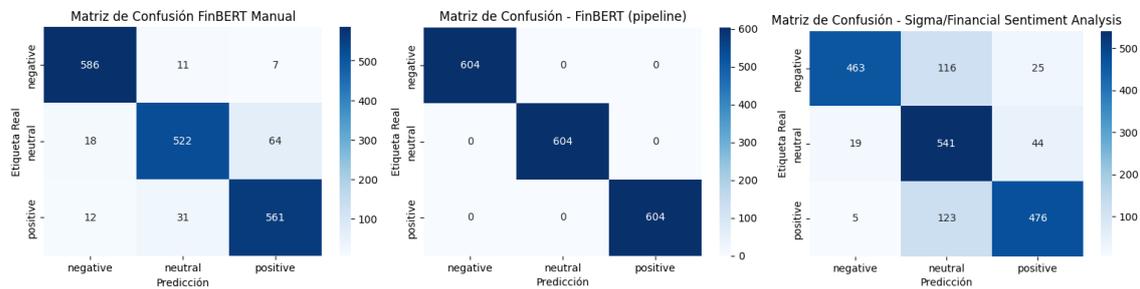


Ilustración 21: Matriz de confusión del enfoque basado en BERT

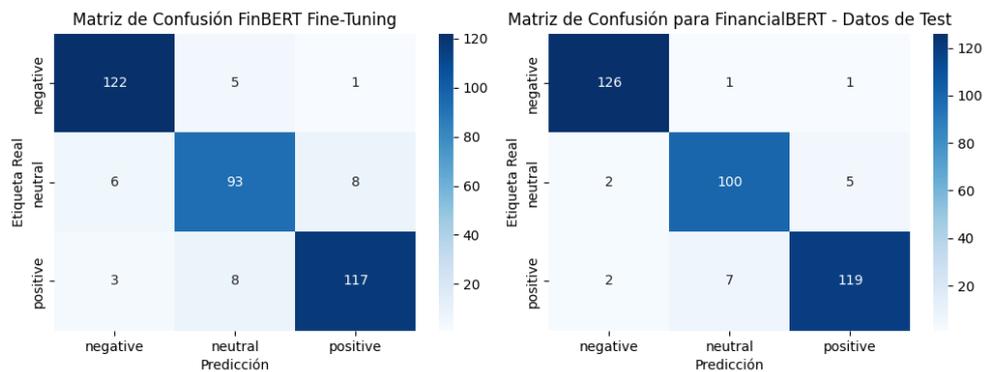


Ilustración 22: Matriz de confusión del enfoque basado en BERT con Fin-Tuning

ENSEMBLE

El modelo de votación mayoritaria supera incluso a los modelos individuales, agregando robustez al incorporar distintas perspectivas. Esto sugiere que la combinación de predicciones ayuda a mitigar errores individuales y ofrece una solución más fiable.

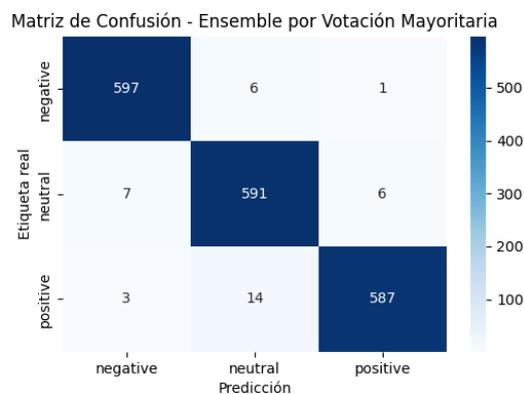


Ilustración 23: Matriz de confusión del enfoque basado en Ensemble

COMPARACIÓN VISUAL

Las matrices de confusión confirman de forma visual las diferencias en rendimiento entre los modelos. En los enfoques basados en transformers, se observa una fuerte concentración de aciertos en la diagonal principal, lo que indica una clasificación precisa y consistente. Por el contrario, los modelos clásicos y los basados en LSTM tienden a mostrar una mayor dispersión, con errores frecuentes entre clases adyacentes, especialmente entre las etiquetas neutral y negative, que resultan más ambiguas. Esta divergencia se alinea con los valores cuantitativos presentados a continuación:

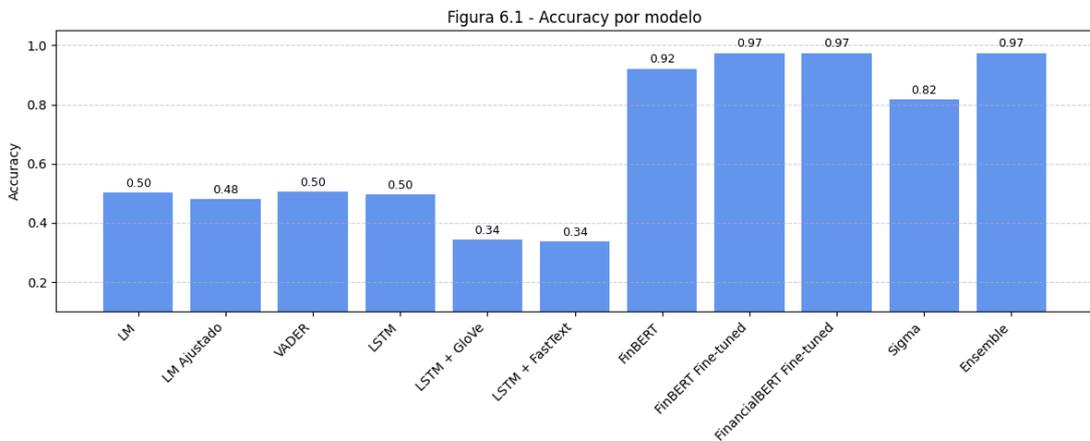


Ilustración 24: Accuracy por modelo de clasificación de sentimiento.

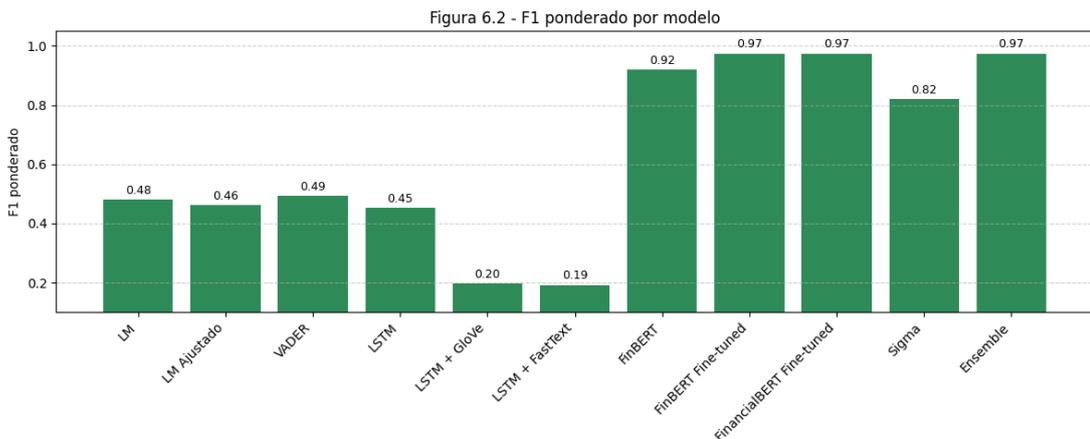


Ilustración 25: F1 ponderado por modelo de clasificación de sentimiento.

Se puede por tanto concluir que la elección del modelo ensemble es la adecuada para la tarea que se ha abordado.

ANÁLISIS DE RESULTADOS DE LA GENERACIÓN DE INFORMES

Este apartado analiza el funcionamiento práctico y el valor aportado por la interfaz desarrollada para la exploración y visualización de resultados. La interfaz implementada permite al usuario acceder fácilmente a diferentes tipos de análisis derivados del procesamiento automático de titulares financieros, combinando visualizaciones intuitivas con opciones interactivas que potencian la interpretación y el descubrimiento de patrones.

DISEÑO DE LA INTERFAZ

La interfaz está organizada de forma modular, permitiendo la navegación libre entre los distintos bloques de análisis mediante un panel lateral. Cada módulo puede utilizarse de manera independiente, lo que favorece una experiencia flexible y orientada a las necesidades del usuario.



Ilustración 26: Ejemplo del Diseño de la Interfaz

Este diseño modular responde a un enfoque centrado en la accesibilidad y la claridad, especialmente pensado para usuarios no técnicos. La separación funcional de los módulos permite explorar diferentes dimensiones del conjunto de datos (contenido textual, sentimiento, temas, mercado) sin sobrecargar la experiencia ni exigir conocimientos avanzados en procesamiento de lenguaje natural o finanzas. Desde el punto de vista visual, se ha priorizado un diseño claro, minimalista e intuitivo, con una jerarquía visual bien definida, una paleta de colores neutra y componentes consistentes a lo largo de toda la interfaz. La estructura se adapta a diferentes tamaños de pantalla y mantiene la información clave en primer plano, sin distracciones ni elementos innecesarios.

La interfaz ha sido diseñada para ser ligera y funcional. En los módulos que requieren mayor carga computacional (como el modelado de tópicos o la generación de resumen automático), el tiempo de espera no supera los 30 segundos en condiciones normales. Durante la carga, se informa visualmente al usuario mediante mensajes de estado, evitando la sensación de bloqueo. En los demás módulos, la carga es prácticamente instantánea.

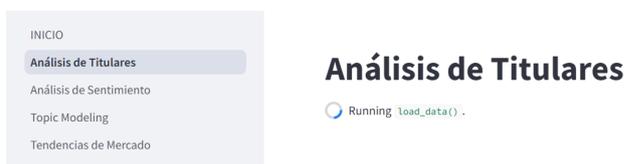


Ilustración 27: Ejemplo de mensaje visual mostrado durante la carga del módulo

Los gráficos utilizados permiten una interacción fluida e intuitiva. Se puede hacer zoom dinámico, acceder a información detallada al pasar el ratón por cada punto (tooltip), y activar opciones de pantalla completa para un análisis más cómodo. Estas funciones mejoran la exploración visual y permiten un análisis más preciso sin recargar la interfaz con elementos adicionales.

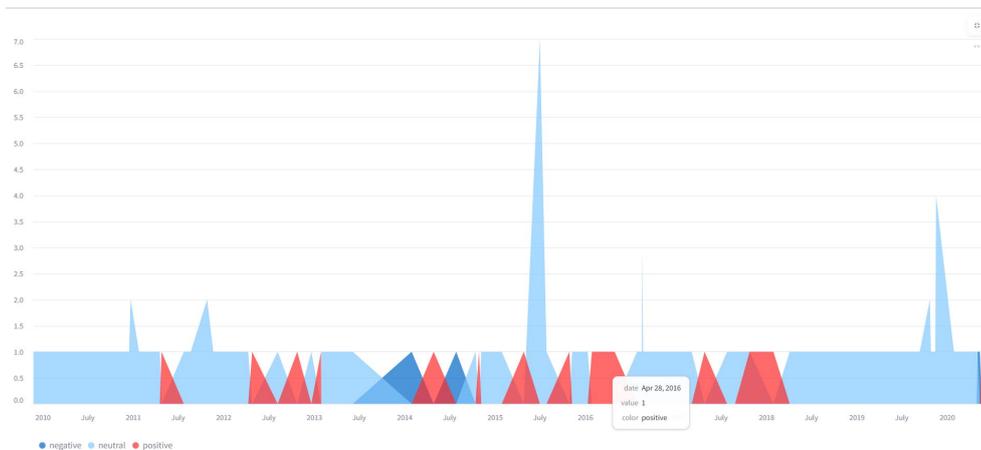


Ilustración 28: Ejemplo de interacción con un gráfico

Asimismo, las opciones de exportación o visualización a pantalla completa añaden valor adicional al uso profesional de la herramienta.



Ilustración 29: Ejemplo de las opción adicionales que ofrecen los gráficos

Se ha incorporado un panel lateral con múltiples filtros personalizables en todos los módulos. El usuario puede seleccionar la empresa, definir un rango de fechas y, en el caso del modelado de tópicos, ajustar el número mínimo de documentos por grupo. Todos los elementos interactivos (botones, sliders, selectores) están dispuestos de forma clara e intuitiva, favoreciendo la exploración progresiva de los resultados.

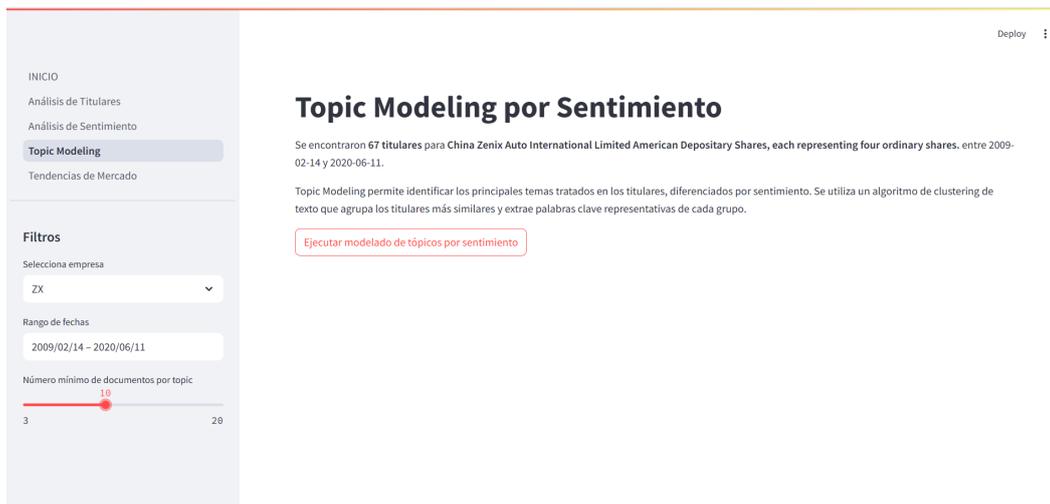


Ilustración 30: Ejemplo del panel lateral de la interfaz

Por último, se han incorporado mensajes de sistema en los puntos clave (por ejemplo, en ausencia de datos suficientes o cuando no se ha seleccionado correctamente la fecha), los cuales han sido diseñados para ser fácilmente distinguibles, sin interrumpir el flujo visual de la página.



Ilustración 31: Ejemplo de un mensaje informativo por pantalla

Esta retroalimentación ayuda al usuario a interpretar correctamente las posibles limitaciones del dataset o de las configuraciones seleccionadas.

FUNCIONAMIENTO Y CARACTERÍSTICAS CLAVE

Para complementar el análisis cuantitativo y la evaluación visual de la herramienta, se va a mostrar un ejemplo ilustrativo del funcionamiento completo de la aplicación, recorriendo sus distintos módulos. Mientras que en el [Capítulo 5](#) se presentaron ejemplos correspondientes a la empresa Apple en el periodo 2019-2020, en este apartado se utilizarán los titulares asociados a Tesla Motors entre los años 2015 y 2020 para generar un informe completo y analizar su flujo de información.

A través de este ejemplo, se detallará cómo un usuario puede seleccionar una empresa y un rango temporal, aplicar los filtros disponibles y obtener resultados integrados en forma de gráficos, análisis de sentimiento, nube de palabras, detección de tópicos y resúmenes generados de forma automática. De este modo, se ilustra de manera práctica el valor añadido de la aplicación como herramienta de toma de decisiones. La estructura del código, así como los módulos utilizados, se describen en el Anexo D: Código y estructura de Streamlit.

Análisis de Titulares

El primer módulo se trata del Análisis de Titulares, donde se visualiza el conjunto de noticias asociadas a Tesla entre el 2 de enero de 2015 y el 11 de junio de 2020. En este caso, se encontraron un total de 1930 titulares, que son analizados para extraer las palabras clave más representativas. Esta información se puede encontrar en la cabecera de la interfaz

El módulo genera una nube de palabras clave, donde el tamaño de cada término refleja su frecuencia relativa. Palabras como *tesla*, *stock*, *say*, *model*, *price*, *musk* o *market* destacan por su alta recurrencia, lo que evidencia el peso que tienen dentro del conjunto de noticias.

Además, se incluye un gráfico de barras con la frecuencia de las 25 palabras más comunes. Este tipo de análisis resulta útil para identificar rápidamente los temas más relevantes. En concreto, las 10 palabras más frecuentes son: tesla (1310), stock (297), say (223), model (215), share (209), price (207), market (203), musk (198), china (189) y target (182).

La Ilustración 32 muestra un ejemplo visual de la interfaz descrita. En ella puede observarse cada componente, como la nube de palabras o el gráfico de frecuencias, que se actualizan de forma dinámica en función de los filtros aplicados en el módulo.

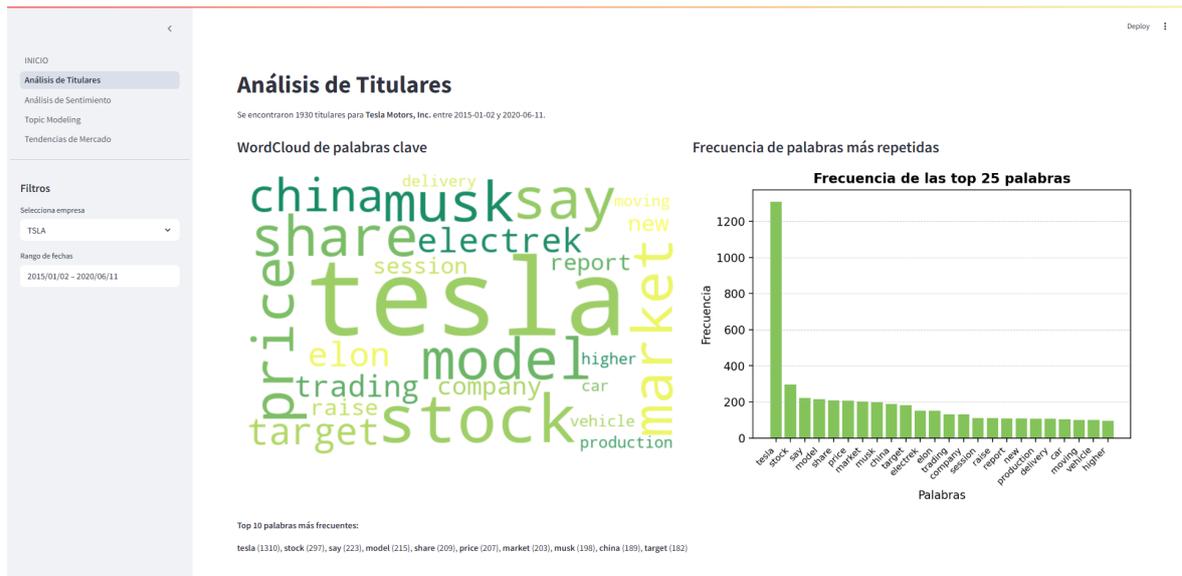


Ilustración 32: Análisis de Titulares de Tesla (2015-2020)

Análisis de Sentimientos

El segundo módulo se trata del Análisis de Sentimiento, en el que se mantienen los filtros previamente seleccionados.

En primer lugar, se ofrece una distribución general de sentimientos mediante dos visualizaciones complementarias: un gráfico de barras y un gráfico circular. En este caso, el análisis muestra que un 85,7 % de los titulares fueron clasificados como neutrales, frente a un 8,9 % positivos y 5,4 % negativos. Esto indica una cobertura mayormente descriptiva o informativa de la empresa en el periodo analizado.

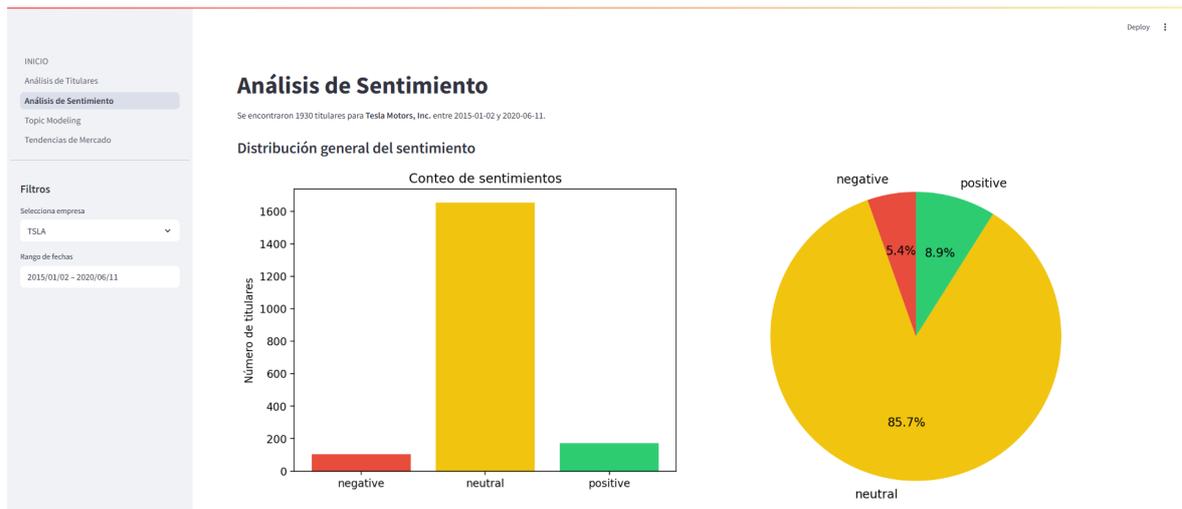


Ilustración 33: Distribución general del sentimiento de Tesla (2015-2020)

A continuación, se presenta una gráfica de evolución temporal que permite visualizar cómo varió la cantidad de titulares de cada clase a lo largo del tiempo. Se observan picos de sentimiento negativo y positivo en momentos puntuales, aunque el volumen de titulares neutrales domina claramente.



Ilustración 34: Evolución temporal del sentimiento de Tesla (2015-2020)

Finalmente, el módulo muestra el sentimiento medio diario, representado mediante una línea continua que permite identificar las fluctuaciones emocionales en torno a la empresa. Este análisis se completa con dos tablas que resumen los 5 días con sentimiento medio más positivo y más negativo.



Ilustración 35: Sentimiento medio diario de Tesla (2015-2020)

Topic Modeling

El tercer módulo de la aplicación permite realizar un análisis más profundo mediante modelado de tópicos, diferenciando los temas más recurrentes en función del sentimiento detectado.

Inicialmente, el usuario debe aplicar los filtros correspondientes para la generación de tópicos. En este caso, se ha seleccionado Tesla entre 2015 y 2020, con un mínimo de 15 documentos por grupo. Una vez ajustados los parámetros, el usuario debe presionar el botón “Ejecutar modelado de tópicos por sentimiento” para iniciar el proceso de clustering.



Ilustración 36: Interfaz de inicio de Topic Modeling

El sistema aplica entonces un algoritmo de agrupamiento sobre los titulares previamente clasificados por sentimiento (positivo, negativo o neutral) y genera tópicos diferenciados por dicha clase. En este ejemplo, se han detectado:

- 5 tópicos positivos
- 15 tópicos neutrales
- 2 tópicos negativos

Cada tópico se presenta con una nube de palabras que refleja los términos más representativos del grupo, así como un resumen en lenguaje natural que sintetiza el contenido principal del clúster. Por ejemplo:

Sentimiento: Positive

Se analizaron 172 documentos para sentimiento positive.

TOPIC 0

Resumen generado: Traders and investors alike are comparing the price of a stock to the price of a share, based on the latest market data.



TOPIC 4

Resumen generado: Share price reported higher in the first quarter of the year compared to the same period last year



Ilustración 37: Generación de tópicos con sentimiento positivo

Sentimiento: Neutral

Se analizaron 1654 documentos para sentimiento neutral.

TOPIC 0

Resumen generado: Tesla says it is raising its target price to a record high of £20 million, but maintains it is not a safe bet.



TOPIC 13

Resumen generado: Investing in technology and investing in the world



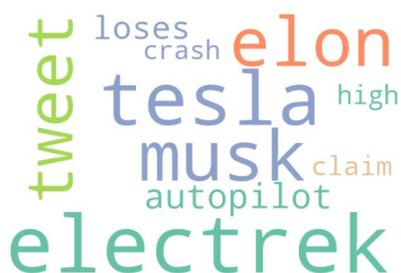
Ilustración 38: Generación de tópicos con sentimiento neutral

Sentimiento: Negative

Se analizaron 104 documentos para sentimiento negative.

TOPIC 0

Resumen generado: Tesla loses claim to have lost control of its autopilot after crash



TOPIC 1

Resumen generado: Tesla shares fall on target delivery in china



Ilustración 39: Generación de tópicos con sentimiento negativo

Estas visualizaciones permiten identificar los temas más relevantes para cada tipo de sentimiento, facilitando una lectura estructurada y comprensible del discurso financiero.

Tendencias de Mercado

El módulo final permite cruzar la información de los titulares analizados con la evolución del mercado financiero. En este caso, para Tesla y el rango de fechas entre el 2 de enero de 2015 y el 10 de junio de 2020, se obtiene una visualización completa de las tendencias bursátiles asociadas.

La primera sección muestra la evolución del precio de cierre diario, representado como una línea temporal. Se observa un crecimiento progresivo del valor de las acciones, con una subida destacable a partir de finales de 2019. A esto se le suma un gráfico de volumen diario, que refleja el nivel de actividad del mercado, mostrando picos significativos en fechas concretas. A su vez, se incluye una gráfica del sentimiento medio diario, que permite observar de forma más detallada la fluctuación del tono general de los titulares a lo largo del tiempo. Esta información añade una capa interpretativa adicional que enriquece el análisis conjunto entre sentimiento y evolución bursátil.

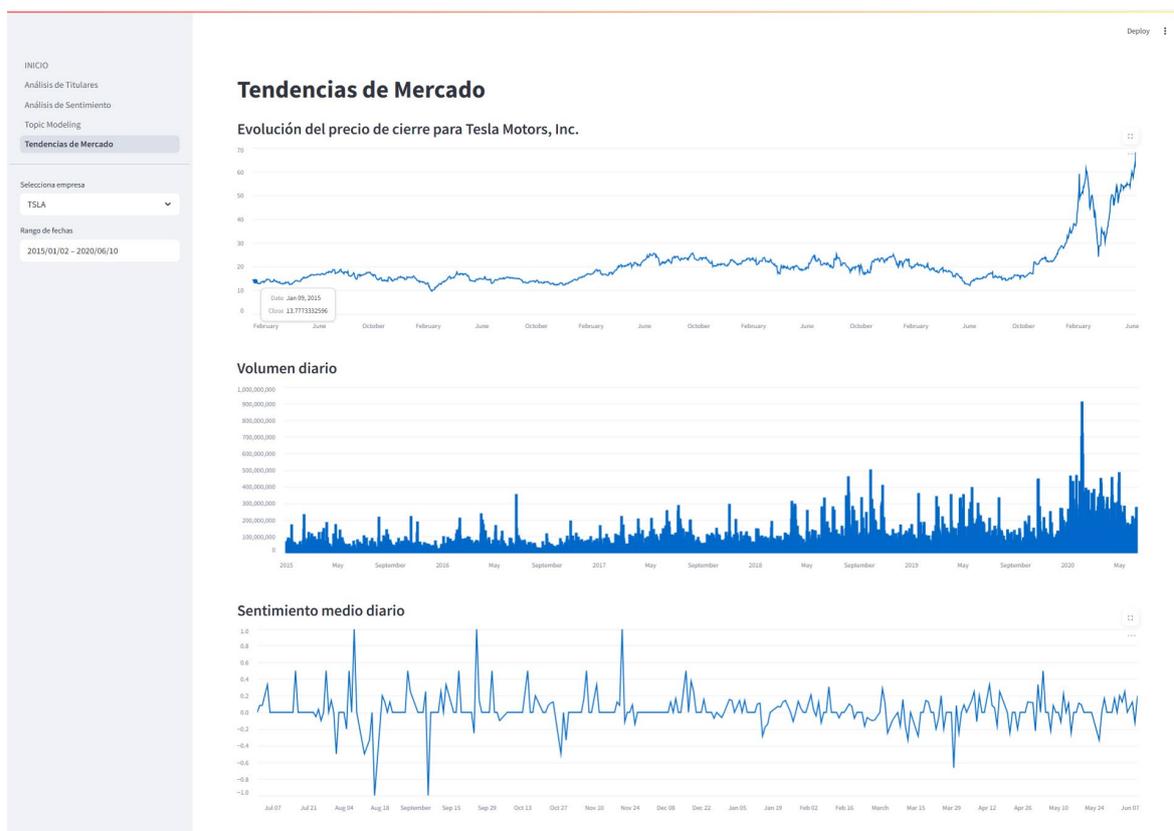


Ilustración 40: Interfaz de Tendencias de Mercado de Tesla (2015-2020)

Se ha incorporado una comparación directa entre el sentimiento medio diario extraído de los titulares y el comportamiento del precio de cierre. La superposición de ambas curvas permite estudiar posibles correlaciones entre el tono de las noticias y las reacciones del mercado. Esta funcionalidad resulta especialmente útil para evaluar el impacto emocional de la narrativa mediática sobre la percepción bursátil.

Por último, el módulo cierra con un resumen generado a partir del conjunto de titulares. En este ejemplo, el sistema identifica como mensaje principal: *"Tesla's stocks have risen to all-time highs after investors raised their price target on the electric vehicle maker."* Este resumen sintetiza la información de los titulares y ofrece al usuario una interpretación final del periodo seleccionado.

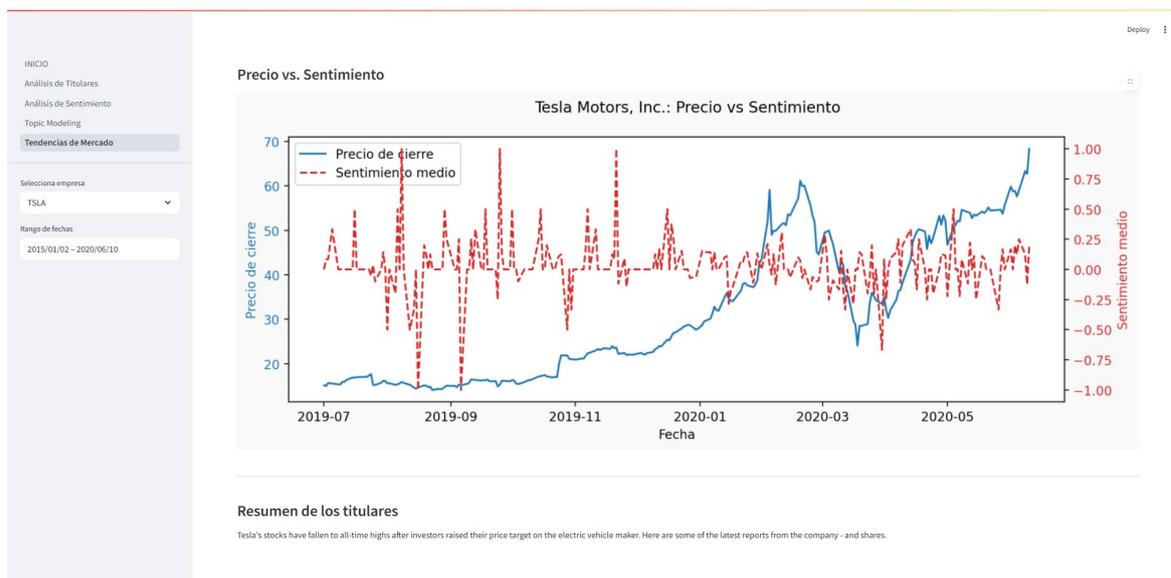


Ilustración 41: Cierre de la Interfaz de Tendencias de Mercado

VALORACIÓN GLOBAL DEL SISTEMA

El sistema desarrollado cumple satisfactoriamente con el objetivo inicial de facilitar el análisis financiero a partir de texto mediante una interfaz accesible, modular y visualmente clara. Cada módulo está enfocado en una tarea concreta, desde la exploración de titulares hasta el cruce entre sentimiento y precios de mercado, permitiendo una interpretación progresiva y comprensible.

La separación funcional entre módulos, junto con la posibilidad de ajustar filtros dinámicos y visualizar los resultados de manera interactiva, ha demostrado ser especialmente útil en contextos exploratorios. Además, la inclusión de modelos de lenguaje avanzados (como FinBERT y FinancialBERT) ha permitido aumentar considerablemente la calidad analítica del sistema.

Desde el punto de vista técnico, el sistema es estable y responde en tiempos razonables. El procesamiento más costoso se realiza de forma previa, por lo que la experiencia de

navegación no se ve penalizada por cálculos en tiempo real. Las visualizaciones, enriquecidas con elementos interactivos, mejoran la comprensión de patrones complejos sin necesidad de recurrir a análisis estadísticos manuales.

En conjunto, la aplicación no solo centraliza el análisis de sentimiento, sino que aporta una capa visual, interpretativa y contextual, que amplía el potencial de uso del dataset original y lo acerca a entornos reales de decisión financiera.

Capítulo 7. CONCLUSIONES FINALES

El desarrollo de este Trabajo de Fin de Máster ha permitido explorar en profundidad las capacidades del análisis de sentimientos aplicado al ámbito financiero, combinando modelos de NLP con técnicas de aprendizaje profundo, generación de informes automáticos y visualización interactiva mediante una interfaz diseñada desde cero. A través de un enfoque iterativo y progresivo, se han comparado enfoques clásicos y modernos para la clasificación de titulares financieros, se han entrenado modelos especializados mediante *fine-tuning*, y se ha propuesto un sistema ensemble capaz de aumentar la robustez de las predicciones al combinar las salidas de distintos modelos preentrenados.

La aplicación final, desarrollada íntegramente en Streamlit, integra los distintos módulos implementados de forma coherente y permite al usuario explorar los datos de forma visual, filtrando por compañía y período temporal. El sistema proporciona análisis detallados sobre la distribución del sentimiento, su evolución temporal, la relación con el comportamiento bursátil, la detección de temas predominantes mediante *topic modeling*, y un resumen automatizado del contenido de los titulares. Esta integración demuestra que **es posible transformar grandes volúmenes de información textual no estructurada en estructurada útil para la toma de decisiones financieras.**

En cuanto al rendimiento de los modelos, se ha observado que los enfoques más simples (como los diccionarios financieros) ofrecen resultados rápidos y explicables, pero carecen de precisión en contextos complejos. Los modelos LSTM mejoran la captura de relaciones secuenciales, pero su efectividad depende en gran medida de los embeddings utilizados. Por otro lado, los modelos tipo Transformer, en especial FinBERT y FinancialBERT, han demostrado un rendimiento excelente, especialmente tras el *fine-tuning*. El sistema ensemble propuesto ha logrado combinar lo mejor de cada uno, incrementando la estabilidad y generalización de las predicciones.

En términos de desarrollo y despliegue de la aplicación, Streamlit ha resultado una herramienta extremadamente útil por su sencillez de desarrollo, su compatibilidad con Python y su capacidad para integrar de forma directa los módulos entrenados previamente en *notebooks*. La estructura modular de la aplicación, dividida en distintas páginas funcionales (*sentiment analysis*, *topic modeling*, *market trends* y resumen), permite un análisis exhaustivo desde distintas perspectivas, manteniendo una experiencia de usuario clara y segmentada.

TRABAJOS FUTUROS

Aunque los resultados obtenidos cumplen con los objetivos iniciales y la aplicación final permite una visualización detallada de distintos enfoques financieros, existen varias líneas de mejora que podrían abordarse en el futuro para ampliar el alcance del proyecto:

RECUPERACIÓN DE DATOS DE MERCADO PARA TICKERS PROBLEMÁTICOS

Durante el desarrollo del proyecto se intentó enriquecer el dataset principal con información sobre el comportamiento bursátil de cada compañía, obtenida a través de la librería `yfinance`, que accede a datos históricos de precios mediante la API de *Yahoo Finance*. El objetivo era disponer de series temporales de cotización para cada *ticker* mencionado en los titulares, lo que permitiría analizar la relación directa entre el sentimiento detectado y la evolución real del mercado. Si bien se logró recuperar datos para una parte significativa del conjunto, numerosos *tickers* presentaron problemas de descarga debido a inconsistencias en los nombres, errores de acceso o falta de datos en la plataforma. Al final, se optó por excluir dichos registros del análisis. Como línea de trabajo futura, sería interesante recuperar la información correspondiente a estos *tickers* problemáticos mediante métodos alternativos (como bases de datos financieras privadas, APIs más robustas o emparejamiento manual), con el fin de obtener una cobertura completa del análisis.

APLICACIÓN DE MODELOS LLM PARA ANÁLISIS DE SENTIMIENTO

Aunque el sistema desarrollado se ha basado principalmente en modelos como FinBERT, FinancialBERT y Sigma (todos ellos altamente especializados en clasificación de sentimiento financiero), estos enfoques se centran en tareas supervisadas estáticas y no aprovechan las capacidades conversacionales ni de razonamiento más complejo que ofrecen los LLMs generalistas como GPT-4, Claude o Gemini.

Una posible ampliación del sistema sería incorporar modelos LLM con capacidad para realizar análisis de sentimiento enriquecido, incluyendo explicaciones del sentimiento detectado, justificación con fragmentos del texto, identificación de ambigüedades o ironía, y generación de comentarios naturales que acompañen las etiquetas. Además, los LLM podrían ser adaptados a tareas auxiliares como la clasificación multitarea (sentimiento + tipo de evento económico) o la extracción de señales de predicción implícita en titulares.

Sin embargo, su despliegue requiere entornos de ejecución con acceso a GPU o servicios cloud avanzados, lo cual no ha sido posible durante el desarrollo del trabajo debido a las limitaciones del entorno de desarrollo. A futuro, la integración de estos modelos permitiría dotar al sistema de un nivel de análisis contextual muy superior al actual.

SIMULACIÓN DE LA EVOLUCIÓN DEL MERCADO BASADA EN SENTIMIENTO

A su vez, se propone diseñar un simulador que, a partir del sentimiento agregado y la evolución temporal de una empresa, estime escenarios futuros de comportamiento del mercado o de acciones específicas. Esto implicaría modelar la relación dinámica entre sentimiento y precios mediante series temporales, modelos causales o incluso entornos de *reinforcement learning* simulados. Además, permitiría evaluar estrategias hipotéticas de inversión y evaluar la utilidad del sentimiento como indicador adelantado, lo cual acercaría aún más el sistema a aplicaciones reales en el ámbito financiero profesional.

Capítulo 8. BIBLIOGRAFÍA

- [1] PwC. (2025). *Medios de pago 2025: más allá del efectivo*. Recuperado de: <https://www.pwc.es/es/financiero/medios-pago-2025-mas-alla.html#dondeestamos>
- [2] Sanger, W., & Warin, T. (2016). *High frequency and unstructured data in finance: An exploratory study of Twitter*. *Journal of Global Research in Computer Science*, 7(4), 1–5. Recuperado de: <https://www.rroij.com/open-access/high-frequency-and-unstructured-data-in-finance-an-exploratory-study-oftwitter.php?aid=70514>
- [3] Marengo, F. (2025, 27 de enero). *FinTech: La Revolución que está Transformando la Industria Financiera*. Recuperado de: <https://www.aden.org/business-magazine/fintech-america-latina-revolucion-financiera>
- [4] Torres, F. (s.f.). *¿Qué es el trading de alta frecuencia y cómo funciona?*. Recuperado de: <https://www.ig.com/es/estrategias-de-trading/-que-es-el-trading-de-alta-frecuencia-y-como-funciona--230616>
- [5] Gradient Team. (2024, 14 de noviembre). *5 Min Rundown: The Role of Unstructured Data in Financial Services*. Recuperado de: <https://www.gradient.ai/blog/5-min-rundown-the-role-of-unstructured-data-in-financial-services>
- [6] Velluti, A. (2024, 5 de noviembre). *The rise of unstructured data in investment: A revolution for decision-maker*. Recuperado de: <https://www.maddyness.com/uk/2024/11/05/the-rise-of-unstructured-data-in-investment-a-revolution-for-decision-maker/>
- [7] Stryker, C., & Holdsworth, J. (2024, 11 de agosto). *What is NLP (natural language processing)?*. Recuperado de: <https://www.ibm.com/think/topics/natural-language-processing>
- [8] Young, T., Hazarika, D., Poria, S. & Cambria, E. (2024, 15 de marzo). *Recent Trends in Deep Learning Based Natural Language Processing*. Recuperado de: <https://arxiv.labs.arxiv.org/html/1708.02709>

- [9] Hirschberg, J. & D. Manning, C. (2015, julio). *Advances in natural language processing*. Recuperado de: https://www.researchgate.net/publication/280117537_Advances_in_natural_language_processing
- [10] M. Santos, L. (2024, 24 de octubre). *Exploring the State of Natural Language Processing: Challenges and Future Directions*. Recuperado de: <https://medium.com/@lorenamelogr/exploring-the-state-of-natural-language-processing-challenges-and-future-directions-e5dacc2cf585>
- [11] Saadani, T. (2022, 29 de septiembre). *NLP use cases in Finance*. Recuperado de: <https://medium.com/ubiai-nlp/nlp-use-cases-in-finance-96adf26a6ea2>
- [12] Need.ai, (2023, 20 de mayo) *Top 9 Applications of NLP in Finance*. Recuperado de: <https://www.needi.ai/blog/top-8-applications-of-nlp-in-finance>
- [13] N. Saret, J & Mitra, S. (2016, mayo). *An AI Approach to Fed Watching*. Recuperado de: https://www.twosigma.com/wp-content/uploads/Ai_approach.pdf
- [14] Bloomberg Professional Services. (2013, 30 de marzo). *Introducing BloombergGPT, Bloomberg's 50-billion parameter large language model, purpose-built from scratch for finance*. Recuperado de: <https://www.bloomberg.com/company/press/bloomberggpt-50-billion-parameter-llm-tuned-finance>
- [15] Charles, E., Iseal, S., Olusegun, J., Henry, M (2024, 16 de noviembre). *Natural language processing (nlp) for financial text analysis*. Recuperado de: <https://www.researchgate.net/publication/385860012>
- [16] Wang, L., Cheng, Y., Xiang, A., Zhang, J., & Yang, H. (s.f.). *Application of Natural Language Processing in Financial Risk Detection*. Afiliaciones: Washington University in St. Louis, Columbia University, University of Electronic Science and Technology of China, University of Chicago, University of Houston.
- [17] Pungulescu, C., & Stolin, D. (s.f). *Measuring Document Similarity: A Comparative Analysis of NLP Methods in Finance*. John Cabot University & TBS Business School.

-
- [18] Barba, O. (2024, 4 de julio). *Natural Language Processing in banking*. Recuperado de: <https://coinscrapfinance.com/banking-innovation/natural-language-processing-banking/>
- [19] Data Science UA. (2024, 4 de noviembre). *NLP in Finance: Revolutionizing Analysis*. Recuperado de: <https://data-science-ua.com/blog/nlp-in-finance-revolutionizing-analysis/>
- [20] Szymańska-Bobowska, K. (2024, 14 de marzo). *How is NLP transforming finance, FinTech, and banking?* Recuperado de: <https://www.future-processing.com/blog/how-nlp-is-transforming-finance-fintech-and-banking/>
- [21] Arsha P. Joy. (2025, 4 de enero). *Building and Implementing Effective NLP Models with Transformers*. Recuperado de: <https://www.skillcamper.com/blog/building-and-implementing-effective-nlp-models-with-transformers>
- [22] Hewlett Packard Enterprise. (s.f.). *¿Qué es el procesamiento del lenguaje natural?* Recuperado de: <https://www.hpe.com/es/es/what-is/nlp.html>
- [23] Sueppel, R. (2019, 15 de junio). *Natural Language Processing & Financial Markets*. Recuperado de: <https://www.linkedin.com/pulse/natural-language-processing-financial-markets-ralph-sueppel>
- [24] Karl, T. (2023, 17 de agosto). *Why is Python Used for Machine Learning?* Recuperado de: <https://www.newhorizons.com/resources/blog/why-is-python-used-for-machine-learning>
- [25] Ebis Education. (2024, 30 de julio). *Jupyter Notebook: Qué es, Para qué sirve y Cómo usarlo*. Recuperado de: <https://www.ebiseducation.com/jupyter-notebook-que-es-para-que-sirve-y-como-usarlo>
- [26] Microsoft. (s.f.). *Visual Studio: IDE y Editor de código para desarrolladores de software y Teams*. Recuperado de: <https://visualstudio.microsoft.com/es/>
- [27] Pandas Developers. (s.f.). *Pandas: Python Data Analysis Library*. Recuperado de: https://pandas.pydata.org/docs/getting_started/index.html
- [28] NumPy Developers. (s.f.). *NumPy*. Recuperado de <https://numpy.org/>

-
- [29] Weights & Biases. (2024, 14 de diciembre). *Financial-BERT*. Recuperado de: <https://wandb.ai/ahmedrachid/FinancialBERT-Language-Model/reports/Financial-BERT--VmlldzoxMzQwMTgy>
- [30] Genc, Z. (2020, 31 de julio). *finBERT: Financial Sentiment Analysis with BERT*. Recuperado de: <https://medium.com/prosus-ai-tech-blog/finbert-financial-sentiment-analysis-with-bert-b277a3607101>
- [31] NLTK Project. (2024, 19 de agosto). *NLTK: Natural Language Toolkit*. Recuperado de: <https://www.nltk.org/>
- [32] Plotly. (s.f.). *Plotly Python Graphing Library*. Recuperado de: <https://plotly.com/python/>
- [33] Plotly. (s.f.). *Deploy Your Dash App*. Recuperado de: <https://dash.plotly.com/deployment>
- [34] Kaggle. (s.f.). *Plataforma de competiciones y datasets de ciencia de datos*. Recuperado de: <https://www.kaggle.com>
- [35] Takala. (s.f.). *Financial PhraseBank [Dataset]*. Hugging Face. Recuperado de: https://huggingface.co/datasets/takala/financial_phrasebank
- [36] Malo, P., Sinha, A., Takala, P., Korhonen, P. y Wallenius, J. (2014): “*Good debt or bad debt: Detecting semantic orientations in economic texts*.” *Journal of the American Society for Information Science and Technology*. Recuperado de: https://www.researchgate.net/publication/251231107_Good_Debt_or_Bad_Debt_Detectin_g_Semantic_Orientations_in_Economic_Texts
- [37] Anelle, M. (2019). *Daily financial news for 6000+ stocks [Dataset]*. Kaggle. Recuperado de: <https://www.kaggle.com/datasets/miguelaenlle/massive-stock-news-analysis-db-for-nlpbacktests/data>
- [38] Anelle, M. (2018). *Scraping Tools – Benzinga [Repositorio GitHub]*. GitHub. Recuperado de: <https://github.com/miguelanelle/Scraping-Tools-Benzinga>

-
- [39] Ramos, R. (2023). *yfinance: Yahoo! Finance market data downloader* [Librería Python]. GitHub. Recuperado de: <https://github.com/ranaroussi/yfinance>
- [40] Bollen, J., Mao, H., & Zeng, X. (2011). *Twitter mood predicts the stock market*. *Journal of Computational Science*, 2(1), 1-8. Recuperado de: <https://doi.org/10.1016/j.jocs.2010.12.007>
- [41] Elia, F. (2023, 18 de agosto). *Sentiment Analysis Dictionaries*. Recuperado de: <https://www.baeldung.com/cs/sentiment-analysis-dictionaries>
- [42] Loughran, T., & McDonald, B. (2024). *Loughran-McDonald Master Dictionary with Sentiment Word Lists* [Data set]. University of Notre Dame. Recuperado de: <https://sraf.nd.edu/loughranmcdonald-master-dictionary/>
- [43] Loughran, T., & McDonald, B. (2024). *Documentation for the Loughran-McDonald Master Dictionary*. University of Notre Dame. Recuperado de: https://www3.nd.edu/~mcdonald/Word_Lists_files/Documentation_LoughranMcDonald_MasterDictionary.pdf
- [44] Geetha, L. (2023, 28 de febrero). *Vader: A Comprehensive Guide to Sentiment Analysis in Python*. Recuperado de: <https://medium.com/@rslavanyageetha/vader-a-comprehensive-guide-to-sentiment-analysis-in-python-c4f1868b0d2e>
- [45] Srivatsavaya, P. (2023, 24 de julio). *LSTM vs GRU*. Recuperado de: <https://medium.com/@prudhviraju.srivatsavaya/lstm-vs-gru-c1209b8ecb5a>
- [46] Pennington, J., Socher, R., & Manning, C. D. (2014). *GloVe: Global Vectors for Word Representation*. *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 1532–1543. Recuperado de: <https://nlp.stanford.edu/pubs/glove.pdf>
- [47] Pennington, J., Socher, R., & Manning, C. D. (2014). *GloVe: Global Vectors for Word Representation*. Stanford NLP Group. Recuperado: <https://nlp.stanford.edu/projects/glove/>

- [48] Navarro, S. (2014, 12 de abril). *¿Qué es FastText y cómo funciona?*. Recuperado de: <https://keepcoding.io/blog/que-es-fasttext-y-como-funciona/>
- [49] Codificando Bits. (s.f.). *BERT: el inicio de una nueva era en el Natural Language Processing*. Recuperado de: <https://codificandobits.com/blog/bert-en-el-natural-language-processing/>
- [50] Cardona, L. (2023, 7 de septiembre). *Google BERT: qué es, cómo funciona y cómo te afecta (2024)*. Recuperado de: <https://www.cyberclick.es/numerical-blog/google-bert-que-es-como-funciona-y-como-te-afecta>
- [51] Murel, J., & Kavlakoglu, E. (2024, 29 de agosto). *What are masked language models?*. Recuperado de: <https://www.ibm.com/think/topics/masked-language-model>
- [52] Risso Charquero, W. A. (2023, 19 de julio). *Procesamiento del Lenguaje Natural en las Finanzas con FinBERT*. Recuperado de: <https://medium.com/@wadrianrisso/procesamiento-del-lenguaje-natural-en-las-finanzas-con-finbert-c39b2c951c60>
- [53] ProsusAI. (s.f.). *FinBERT*. Hugging Face. Recuperado de: <https://huggingface.co/ProsusAI/finbert>
- [54] Sigma. (s.f.). *financial-sentiment-analysis*. Hugging Face. Recuperado de: <https://huggingface.co/Sigma/financial-sentiment-analysis>
- [55] Ahmedrachid (s.f.). *FinancialBERT*. Hugging Face. Recuperado de: <https://huggingface.co/ahmedrachid/FinancialBERT-Sentiment-Analysis>
- [56] Coursera Staff. (2024, 18 de abril). *What are large language models?*. Recuperado de: <https://www.coursera.org/articles/large-language-models>
- [57] AI4Finance Foundation. (2024). *FinGPT* [Repositorio GitHub]. GitHub. Recuperado de: <https://github.com/AI4Finance-Foundation/FinGPT?tab=readme-ov-file>

- [58] Jain, A. (2024, 13 de septiembre). *A comprehensive guide to ensemble techniques: Bagging and boosting*. Recuperado de: <https://medium.com/@abhishekjainindore24/a-comprehensive-guide-to-ensemble-techniques-bagging-and-boosting-fa276e28da9f>
- [59] Singh, A. (2025, 4 de abril). *A comprehensive guide to ensemble learning (with Python codes)*. Recuperado de: <https://www.analyticsvidhya.com/blog/2018/06/comprehensive-guide-for-ensemble-models/>
- [60] Google. (s.f.). *flan-t5-base*. Hugging Face. Recuperado de: <https://huggingface.co/google/flan-t5-base>
- [61] Csebuetnlp. (s.f.). *mT5-multilingual-XLsum*. Hugging Face. Recuperado de: https://huggingface.co/csebuetnlp/mT5_multilingual_XLSum
- [62] Nantasenamat, C. (2024, 23 de julio). *Streamlit 101: The fundamentals of a Python data app*. Streamlit Blog. Recuperado de: <https://blog.streamlit.io/streamlit-101-python-data-app/>
- [63] Nantasenamat, C. (2024, 22 de enero). *Building a dashboard in Python using Streamlit*. Streamlit Blog. Recuperado de: <https://blog.streamlit.io/crafting-a-dashboard-app-in-python-using-streamlit/>
- [64] Naciones Unidas. (s.f.). *Objetivo 8: Trabajo decente y crecimiento económico*. Recuperado de: <https://www.un.org/sustainabledevelopment/es/economic-growth/>
- [65] Naciones Unidas. (s.f.). *Objetivo 9: Industria, innovación e infraestructura*. Recuperado de: <https://www.un.org/sustainabledevelopment/es/infrastructure/>
- [66] Naciones Unidas. (s.f.). *Objetivo 17: Alianzas para lograr los objetivos*. Recuperado de: <https://www.un.org/sustainabledevelopment/es/globalpartnerships/>
- [67] Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*. In Proceedings of NAACL-HLT 2019 (pp. 4171–4186). Recuperado de: <https://aclanthology.org/N19-1423/>
- [68] FARaci, D. (2019). *FinBERT: Financial Sentiment Analysis with Pre-trained Language Models*. Recuperado de: <https://arxiv.org/abs/2006.08097>

ANEXOS

ANEXO A: ODS

Este proyecto no solo tiene un enfoque técnico, sino que también busca generar un impacto positivo más allá del ámbito académico, alineándose de forma clara con varios Objetivos de Desarrollo Sostenible (ODS) de la Agenda 2030 de Naciones Unidas, en concreto:

ODS 8 – TRABAJO DECENTE Y CRECIMIENTO ECONÓMICO

El desarrollo de la plataforma de análisis financiero contribuye a modernizar el sector mediante la incorporación de tecnologías avanzadas, promoviendo la eficiencia y la innovación en la toma de decisiones económicas [64].

- Facilita la interpretación de grandes volúmenes de información financiera, reduciendo barreras técnicas y mejorando el rendimiento de analistas y profesionales.
- Promueve el crecimiento económico a través del uso de soluciones tecnológicas que optimizan el análisis de mercados.

ODS 9 – INDUSTRIA, INNOVACIÓN E INFRAESTRUCTURA

Este trabajo representa un claro ejemplo de cómo la inteligencia artificial y el aprendizaje profundo pueden integrarse en sectores clave como el financiero [65].

- Se apuesta por la innovación tecnológica mediante el uso de modelos de NLP avanzados, y se construye un sistema para transformar datos no estructurados en información para la toma de decisión.
- La plataforma desarrollada es flexible y escalable, lo que permite su evolución futura hacia nuevos casos de uso o aplicaciones.

ODS 17 – ALIANZAS PARA LOGRAR LOS OBJETIVOS

El proyecto parte del uso de datos abiertos y tecnologías accesibles que facilitan la colaboración entre distintos sectores [66].

- Utiliza herramientas y librerías *open source* como Streamlit, Hugging Face o yfinance, fomentando un enfoque colaborativo y replicable.
- Plantea un puente entre el mundo financiero y el tecnológico, mostrando cómo la combinación de estas disciplinas puede dar lugar a soluciones completas.

ANEXO B: EXTRACCIÓN DE DATOS FINANCIEROS

A lo largo del trabajo se han explicado los pasos empleados de limpieza y transformación de los datos, sin incluir el código correspondiente. Por ello, en esta sección se va a exponer y explicar el *script* completo utilizado para la descarga, depuración y refuerzo de los datos bursátiles desde la API de Yahoo Finance, usando la librería yfinance en Python. Se ha incorporado manejo de errores, control de pausas para evitar bloqueos, y una etapa final de combinación de ficheros CSV en un único dataset.

Inicialmente, se define el rango temporal para la descarga de datos y se crea (si no existe) la carpeta donde se almacenarán los archivos descargados:

```
# Definir el rango de fechas
start_date = "2009-02-14"
end_date = "2020-06-11"

# Ruta de salida
output_folder = "input/stock/stock/data"
os.makedirs(output_folder, exist_ok=True)
```

A partir del dataset principal, el que contiene los titulares sin categorizar, se extraen los *tickers* únicos sobre los que se quiere obtener la información bursátil. A su vez, se inicializa una lista donde se irán guardando los *tickers* que no devuelvan datos válidos.

```
# Lista de tickers a descargar
unique_tickers = df_arp['stock'].unique()

# Lista para almacenar tickers que dan error
error_tickers = []
```

Se inicia un bucle sobre cada *ticker*. Se introduce una pausa de 2 segundos entre descargas para evitar bloqueos o limitaciones por parte de la API. Si no se obtienen datos (el DataFrame está vacío), se guarda el *ticker* en la lista de errores y se pasa al siguiente.

```
for ticker in unique_tickers:
    try:
        print(f"Descargando datos para {ticker} desde Yahoo Finance...")

        # Evitar bloqueos
        time.sleep(2)

        # Descargar datos históricos
        df = yf.download(ticker, start=start_date, end=end_date)

        # DataFrame vacío
        if df.empty:
            print(f"No hay datos para {ticker}. Saltando...")
            error_tickers.append(ticker)
            continue
```

Se reinicia el índice para tener la columna Date como una columna normal, y se añade la columna Company con el *ticker* correspondiente. Luego, se guarda el DataFrame como CSV.

```
# Reseteo índice
df.reset_index(inplace=True)

# Agregar columna "Company" en la primera posición
df.insert(0, "Company", ticker)
output_file = os.path.join(output_folder, f"{ticker}_data.csv")
df.to_csv(output_file, index=False, encoding="utf-8", mode="w",
lineterminator="\n")

print(f"Datos de {ticker} guardados en {output_file}")
```

En algunas ocasiones, la descarga genera una fila vacía o mal formateada en la primera posición. Para asegurar uniformidad, se elimina la primera fila y se guarda de nuevo el archivo limpio.

```
df = pd.read_csv(output_file) # Volver a cargar el archivo
df = df.iloc[1:].reset_index(drop=True) # Eliminar primera fila
df.to_csv(output_file, index=False, encoding="utf-8", mode="w",
lineterminator="\n") # Guardar el archivo limpio

print(f"Archivo limpio guardado: {output_file}")
```

Se ha añadido un bloque try-except para capturar cualquier excepción que pueda surgir y continuar con el siguiente *ticker* sin interrumpir la ejecución.

```
except Exception as e:
    print(f"Error inesperado al descargar datos para {ticker}: {e}")
    error_tickers.append(ticker)
```

Si algún *ticker* falló, se guarda la lista completa en un archivo .txt para su posterior revisión o reintento manual.

```
# Guardar la lista de tickers que han dado error
if error_tickers:
    error_file = os.path.join(output_folder, "tickers_fallidos.txt")
    with open(error_file, "w", encoding="utf-8") as f:
        for ticker in error_tickers:
            f.write(ticker + "\n")
    print(f"Lista de tickers fallidos guardada en {error_file}")
```

Finalmente, se recorren todos los archivos generados en la carpeta de salida, se leen en memoria y se concatenan en un solo DataFrame final que se guarda como `data_stock.csv`, el cual se empleará para el análisis de tendencias de mercado en la aplicación web. Si algún archivo presenta errores de lectura, se muestra un aviso.

ANEXO C: MÉTRICAS DE EVALUACIÓN

Para evaluar el rendimiento de los modelos de clasificación de sentimiento desarrollados a lo largo del proyecto, se han empleado diversas métricas estándar en tareas de clasificación multiclase (positivo, negativo, neutral).

ACCURACY

Indica el porcentaje de predicciones correctas sobre el total de ejemplos evaluados y se caracteriza por su fácil interpretabilidad.

$$Accuracy = \frac{\text{Número de predicciones correctas}}{\text{Total de predicciones}}$$

Es importante no tomar esta medida como única referencia, ya que aunque es útil como una primera instancia, puede llevar a error si las clases están desbalanceadas. Por ejemplo, si la mayoría de los titulares son neutros, un modelo que siempre prediga "neutral" puede tener una *accuracy* alto sin ser realmente bueno.

PRECISION (PRECISIÓN)

Para una clase concreta (por ejemplo, "positivo"), la precisión indica qué proporción de las predicciones que hizo el modelo para esa clase son realmente correctas.

$$Precision_i = \frac{(TP_i)}{TP_i + FP_i}, \quad \begin{cases} TP_i: \text{Verdaderos positivos de la clase } i \\ FP_i: \text{Falsos positivos de la clase } i \end{cases}$$

Es especialmente importante cuando cada predicción errónea puede tener consecuencias significativas. En el contexto financiero, esto es muy relevante. Por ejemplo, si un modelo clasifica erróneamente un titular negativo como positivo, un analista podría interpretar que una empresa está en buena situación cuando en realidad se enfrenta a problemas. Por tanto, una alta precisión reduce el riesgo de emitir señales falsas.

RECALL (SENSIBILIDAD)

Para una clase concreta, mide cuántos de los ejemplos verdaderos de esa clase el modelo fue capaz de identificar correctamente.

$$Recall_i = \frac{TP_i}{TP_i + FN_i} \begin{cases} TP_i: \text{Verdaderos positivos de la clase } i \\ FN_i: \text{Falsos negativos de la clase } i \end{cases}$$

Es especialmente relevante cuando se quieren minimizar los falsos negativos, es decir, cuando se quiere evitar dejar de clasificar como positiva una noticia que en realidad sí lo es. En el ámbito financiero, esto puede marcar una diferencia significativa, ya que no detectar un titular que realmente transmite confianza sobre una empresa podría hacer que se pierdan oportunidades. Por ello, un recall alto es deseable en escenarios donde los errores de tipo 'falso negativo' pueden tener consecuencias económicas o estratégicas importantes

F1 SCORE

Es la media armónica entre *precision* y *recall*. Es una métrica equilibrada que penaliza los extremos (es decir, un modelo con mucha *precision* pero poco *recall*, o al revés, tendrá un F1 bajo).

$$F1_i = 2 \times \frac{(Precision_i \times Recall_i)}{Precision_i + Recall_i}$$

Es especialmente útil cuando hay clases desbalanceadas, como suele ocurrir en el análisis de sentimientos. Se ha empleado como una de las métricas principales de referencia para comparar el rendimiento de los modelos entre clases (positivo, neutro, negativo).

MATRIZ DE CONFUSIÓN

Se trata de una matriz que permite visualizar fácilmente cómo se están clasificando los ejemplos. Las filas indican las clases reales, y las columnas las predichas.

	Pred. Positivo	Pred. Neutro	Pred. Negativo
Real Positivo	TP	FN	FN
Real Neutro	FP	TP	FN
Real Negativo	FP	FN	TP

Ilustración 42: Ejemplo matriz de confusión de 3 clases

Más allá de los valores numéricos, la matriz de confusión muestra de forma visual los errores del modelo. Por ejemplo, si confunde los titulares negativos con neutros. Esto permite ajustar el modelo o incorporar mejoras específicas en el preprocesamiento.

ANEXO D: ESTRUCTURA DE STREAMLIT

La plataforma desarrollada en este proyecto se ha diseñado usando Streamlit. Dado que la interfaz permite navegar por distintos módulos (análisis de sentimiento, generación de resúmenes, análisis de tópicos, y tendencias del mercado), cada uno se ha implementado como una página independiente dentro del directorio /pages, siguiendo una estructura modular que facilita tanto el mantenimiento como la escalabilidad.

```
TFM/
├── .vscode/
├── data/
├── models/
│   ├── financialbert_finetuned/
│   ├── finbert_finetuned/
│   └── sigma/
├── pages/
│   ├── 1_Análisis_de_Titulares.py
│   ├── 2_Análisis_de_Sentimiento.py
│   ├── 3_Topic_Modeling.py
│   └── 4_Tendencias_de_Mercado.py
├── utils/
└── INICIO.py
```

En este anexo se describen las funcionalidades principales de cada página, explicando el flujo del código. La estructura modular permite que cada componente se ejecute de forma independiente. Además, se incorporan distintos modelos preentrenados (y fine-tuneados) de NLP y se integran datos bursátiles, lo que convierte a la plataforma en una herramienta versátil para el análisis financiero.

ANÁLISIS DE TITULARES

Al iniciar la página, se configura el diseño con `st.set_page_config(layout="wide")` para aprovechar al máximo el espacio horizontal. Se carga un dataset previamente procesado (`titulares_sentimiento.csv`) donde cada titular está asociado a una fecha, un *ticker* de empresa y una etiqueta de sentimiento. También se carga un diccionario de mapeo entre los *tickers* y el nombre completo de las empresas desde el archivo `empresas.csv`.

El usuario puede seleccionar una empresa de entre todas las disponibles y definir un rango de fechas. Estos filtros se aplican sobre el DataFrame para generar una vista personalizada. Si no se encuentran titulares dentro del rango, se lanza una advertencia visual.

Los titulares se limpian con una función propia `clean()` que normaliza, tokeniza y filtra las palabras (por ejemplo, eliminando *stopwords*). Luego, se agrupan todas las palabras en una lista y se construye un Counter para contar cuántas veces aparece cada término. Se seleccionan las 25 más frecuentes (ignorando números y palabras muy cortas).

Se presentan dos visualizaciones principales en paralelo:

- **WordCloud:** Una nube de palabras en la que el tamaño de cada término representa su frecuencia relativa. Se genera con la librería `wordcloud` y se adapta a los colores de la app para mantener la coherencia visual.
- **Gráfico de barras:** Muestra la frecuencia exacta de las 25 palabras más comunes en formato barra, con etiquetas rotadas y una cuadrícula para facilitar la lectura.

Estas visualizaciones permiten detectar rápidamente patrones. Finalmente, se genera una breve lista textual con el top 10 de palabras más repetidas junto con su número de frecuencia, destacadas en negrita.

ANÁLISIS DE SENTIMIENTOS

La lógica inicial es idéntica a la página anterior. Primero se carga el dataset con los titulares clasificados, se mapea cada *ticker* a su nombre de empresa y se aplican filtros interactivos para elegir una empresa y un rango de fechas específico. Si no hay titulares en el intervalo, se lanza una advertencia y se detiene la ejecución.

Una vez filtrados los titulares, se genera una doble visualización para mostrar la distribución general de sentimientos:

- **Gráfico de barras:** permite observar el número de titulares clasificados como positivos, neutros o negativos.
- **Gráfico circular (tarta):** muestra la proporción relativa de cada clase, facilitando una lectura más intuitiva del reparto emocional.

Ambos gráficos utilizan una codificación de color consistente: rojo para negativo, amarillo para neutro y verde para positivo.

A continuación, se muestra una gráfica de áreas que representa el número diario de titulares por clase de sentimiento. Esto permite visualizar tendencias y momentos de mayor volumen emocional.

A cada clase de sentimiento se le asigna un valor numérico (-1, 0, +1) y se calcula la media diaria. Esto se representa en un gráfico de línea que muestra cómo varía el “sentimiento medio” con el tiempo. Si la línea sube, predomina el optimismo y si baja, el pesimismo.

Finalmente, se muestran dos tablas con los cinco días más positivos y los cinco más negativos según este valor medio. Esta sección es útil para destacar fechas clave, que el usuario puede después explorar más a fondo en otras secciones de la app.

TOPIC MODELING

Se utiliza el modelo `google/flan-t5-base` de Hugging Face para generar automáticamente un pequeño resumen textual de cada tópico detectado. Esta funcionalidad se carga al inicio y se mantiene en caché para mejorar el rendimiento.

Como se lleva explicando en las dos páginas anteriores, el usuario puede seleccionar la empresa y un rango de fechas concreto. También se permite definir el número mínimo de titulares que debe tener un grupo temático para ser considerado un tópico válido (ajustable mediante un slider lateral).

Cuando se pulsa el botón “Ejecutar modelado de tópicos por sentimiento”, el script procesa los titulares de cada clase emocional (*positive*, *neutral*, *negative*) por separado. Para cada clase:

- Se agrupan los titulares más similares entre sí.
- Se extraen las palabras clave más representativas del grupo.
- Se genera un resumen del contenido general de ese tópico .
- Se muestra una nube de palabras por cada tópico para facilitar la interpretación visual.

En caso de que una clase de sentimiento no tenga suficientes titulares o no se generen tópicos relevantes, se muestra un mensaje por pantalla para evitar confusión.

TENDENCIAS DE MERCADO

Como en las demás páginas, se permite seleccionar la empresa y el rango de fechas. Una vez aplicados los filtros, se combinan dos datasets: el que contiene los sentimientos diarios agregados y el de precios históricos de acciones.

Se genera una línea temporal que representa la evolución del precio de cierre diario de la empresa seleccionada. Este gráfico es clave para identificar posibles correlaciones visuales entre cambios en el sentimiento y fluctuaciones del precio.

Debajo del gráfico de precios se presenta un gráfico de barras que muestra el volumen diario de operaciones. Los picos de volumen suelen coincidir con eventos relevantes o periodos de alta volatilidad, por lo que resulta útil compararlos con el análisis emocional previo.

Se incluye una visualización del sentimiento medio diario en forma de línea. Este *score* se obtiene a partir de la puntuación de cada titular y permite observar cómo ha evolucionado la percepción general del mercado sobre la empresa.

Finalmente, se ofrece la posibilidad de generar un resumen en lenguaje natural de los principales titulares de ese periodo usando el modelo `csebuetnlp/mT5_multilingual_XLSum`. Esta funcionalidad resume los *insights* clave del periodo seleccionado y refuerza el valor explicativo de la plataforma.