Análisis de sentimientos en noticias financieras y su impacto en la bolsa de valores de Colombia – COLCAP

Elaborado por:

Luis Fernando Cruz Montaña

Luisa Fernanda Tirado León

Universidad Ean

Escuela de Formación en Investigación

Seminario de Investigación de Especialización

Bogotá

01/06/2025

Resumen

El presente informe aborda el análisis del sentimiento en noticias financieras y su impacto sobre el índice COLCAP. Se propone aplicar técnicas de Procesamiento de Lenguaje Natural para explorar la relación entre la percepción mediática y el comportamiento bursátil en Colombia. El estudio se sustenta en literatura reciente y busca aportar un modelo replicable. Además, se contemplan métodos de aprendizaje automático supervisado y no supervisado para clasificar el tono de las noticias, evaluando su correlación con las variaciones del índice. Esta investigación pretende ofrecer herramientas útiles para la toma de decisiones en inversión y gestión de riesgos.

Palabras clave: análisis de sentimientos, COLCAP, NLP, noticias financieras, mercado bursátil, aprendizaje automático.

Problema de investigación

La metodología de análisis de sentimientos se ha convertido en una herramienta relevante en el ámbito financiero, permitiendo evaluar cómo las emociones y percepciones del mercado afectan los precios de activos financieros. Estudios como los de Bollen, Mao y Zeng (2011) y Shynkevich et al. (2016) han demostrado que el sentimiento extraído de fuentes como redes sociales o noticias financieras tiene una influencia significativa en la volatilidad y rendimiento de índices bursátiles como el S&P 500 y el Dow Jones. Asimismo, Tetlock (2007) evidenció que el tono de las noticias económicas tiene un impacto estadísticamente significativo sobre los rendimientos del mercado, mientras que Loughran y McDonald (2011) desarrollaron diccionarios específicos para el análisis financiero que mejoran la precisión del análisis de sentimientos en este dominio.

En el contexto colombiano, particularmente respecto al índice COLCAP, existe una escasa aplicación de técnicas de procesamiento de lenguaje natural (NLP) y aprendizaje

automático para evaluar el impacto del sentimiento en su comportamiento. Investigaciones como las de García (2013) sugieren que los mercados emergentes, como el colombiano, son especialmente sensibles a los flujos de información y al sentimiento del inversor. No obstante, aún falta evidencia empírica que vincule directamente la percepción mediática con el comportamiento del COLCAP.

El COLCAP representa un indicador fundamental para los inversionistas locales, reflejando el desempeño de las principales acciones listadas en la Bolsa de Valores de Colombia. No obstante, la toma de decisiones en el mercado financiero no se basa únicamente en indicadores económicos y financieros objetivos, sino que también está influenciada por factores psicológicos y percepciones del mercado (Barberis, Shleifer & Vishny, 1998; Kahneman & Tversky, 1979). Esta sensibilidad se acentúa en contextos de alta incertidumbre económica y política, donde la narrativa y el lenguaje utilizado por los medios pueden amplificar reacciones del mercado (Shiller, 2017).

En este sentido, el crecimiento exponencial de fuentes informativas digitales, como medios de comunicación económicos y redes sociales, plantea una oportunidad para estudiar cómo la percepción mediática puede influenciar las decisiones de inversión. La carencia de investigaciones aplicadas al contexto colombiano abre la posibilidad de contribuir tanto a nivel académico como práctico con herramientas que ayuden a anticipar movimientos del mercado. Este proyecto, por tanto, busca cerrar dicha brecha mediante el uso de modelos de aprendizaje automático que permitan analizar el sentimiento del mercado en relación con el COLCAP.

Pregunta de investigación:

¿Cuál es la tendencia del sentimiento en las noticias financieras y cómo se relaciona con las variaciones del índice COLCAP en los últimos 3 años

Objetivos

Objetivo general

Analizar la tendencia del sentimiento en las noticias financieras mediante técnicas de Procesamiento de Lenguaje Natural (NLP) y evaluar su relación con las variaciones del índice COLCAP en los últimos tres años.

Objetivos específicos

1. Recolectar y preprocesar un conjunto de datos de noticias financieras relacionadas con el mercado colombiano durante los últimos tres años.
2. Aplicar técnicas de Procesamiento de Lenguaje Natural (NLP) para extraer y clasificar el sentimiento de las noticias financieras.
3. Determinar la tendencia temporal del sentimiento en las noticias financieras utilizando métodos de análisis de series de tiempo.
4. Evaluar la correlación entre el sentimiento de las noticias y las variaciones del índice COLCAP mediante técnicas estadísticas y de machine learning.
5. Explorar la capacidad predictiva del análisis de sentimientos sobre los movimientos del COLCAP, considerando distintos enfoques de NLP.
6. Interpretar los resultados y proponer recomendaciones sobre el uso del análisis de sentimientos en la toma de decisiones financieras.

Justificación

El análisis de sentimientos aplicado a noticias financieras representa una oportunidad valiosa para comprender la influencia que ejercen las emociones colectivas y la percepción mediática en los mercados bursátiles. En Colombia, el índice COLCAP es el principal referente del comportamiento del mercado accionario; sin embargo, la investigación académica sobre

cómo factores cualitativos como el sentimiento mediático inciden en su variación aún es incipiente. En este contexto, el estudio resulta conveniente al proponer un modelo de aplicación que incorpora técnicas de Procesamiento de Lenguaje Natural (NLP) y aprendizaje automático, alineándose con el enfoque tecnológico y aplicado del programa de especialización en Machine Learning.

Desde una perspectiva de relevancia social, este proyecto puede contribuir a la toma de decisiones más informadas por parte de inversionistas, analistas financieros y gestores de portafolio, quienes enfrentan entornos de alta incertidumbre económica y necesitan herramientas que integren variables cuantitativas y cualitativas. La investigación también posee implicaciones prácticas significativas, al ofrecer un modelo replicable que permite monitorear en tiempo real el sentimiento en el ecosistema financiero colombiano, optimizando así la respuesta del mercado ante eventos mediáticos.

En cuanto al valor teórico, este trabajo aporta al cuerpo de conocimiento existente al abordar una brecha específica en la literatura nacional sobre la interrelación entre análisis de sentimientos y comportamiento bursátil. Complementa hallazgos internacionales al adaptarlos a las condiciones del mercado colombiano, lo cual enriquece el debate académico sobre la integración entre finanzas, inteligencia artificial y ciencias del comportamiento.

La utilidad metodológica del estudio radica en el diseño de un enfoque híbrido que combina análisis textual, minería de datos y modelos predictivos, permitiendo no solo replicar la metodología en otros mercados, sino también ser adaptada a distintos tipos de contenido textual como reportes empresariales o redes sociales.

Este proyecto se enmarca dentro del Campo de Ciencia, Tecnología e Innovación, en el Grupo de Ciencias Básicas, y en la Línea de investigación en Inteligencia Artificial y Análisis de Datos, de acuerdo con los lineamientos establecidos por la Universidad EAN. De esta manera, el estudio contribuye a los propósitos institucionales de fomentar investigaciones aplicadas, con

impacto real en el entorno empresarial y alineadas con las tendencias tecnológicas contemporáneas.

Marco Teórico

El análisis de sentimientos, una rama del Procesamiento de Lenguaje Natural (PLN), se centra en identificar y extraer opiniones, emociones y actitudes expresadas en datos textuales. En el ámbito financiero, esta técnica se ha utilizado para evaluar cómo las percepciones del mercado influyen en las variaciones de precios de los activos financieros. Estudios previos han demostrado que el sentimiento derivado de fuentes como noticias financieras y redes sociales puede tener un impacto significativo en la volatilidad y el rendimiento de los mercados bursátiles (Bollen, Mao & Zeng, 2011; Tetlock, 2007).

El análisis de sentimientos se basa en la premisa de que el lenguaje natural contiene señales afectivas que, cuando se analizan sistemáticamente, pueden proporcionar información valiosa sobre las expectativas del mercado. Investigaciones recientes han explorado la relación entre el sentimiento del mercado y el comportamiento de los índices bursátiles. Por ejemplo, un estudio bibliométrico analizó 223 artículos relacionados con el análisis de sentimientos en la investigación del mercado de valores, destacando la creciente importancia de esta técnica en el ámbito financiero (Hafez et al., 2023). Otro estudio implementó el modelo FinBERT, una variante de BERT ajustada para textos financieros, para analizar el sentimiento de los titulares de noticias del mercado de valores, demostrando la eficacia de los modelos de lenguaje preentrenados en este dominio (Araci, 2019).

El uso de PLN para predecir tendencias en el mercado de valores ha ganado atención en la comunidad académica. Zhang et al. (2018) proporcionaron evidencia sobre cómo utilizar PLN para mejorar la predicción de precios de acciones, mostrando una correlación significativa entre los titulares de noticias y las fluctuaciones de precios. Además, el análisis de sentimientos

ha demostrado ser eficaz para anticipar tendencias en condiciones de alta volatilidad (Kordonis et al., 2016).

La integración de modelos de aprendizaje automático en el análisis financiero ha permitido avances significativos. Un artículo de revisión elaborado por Sun et al. (2022) exploró el uso de herramientas de análisis de sentimientos en la predicción de la volatilidad del mercado bursátil, enfocándose en enfoques de inteligencia artificial, minería de texto y aprendizaje profundo. Asimismo, Chen et al. (2020) utilizaron técnicas de PLN para evaluar el sentimiento del mercado a partir de artículos de noticias, informes financieros y publicaciones en redes sociales, con el objetivo de comprender su impacto en los precios de las acciones.

Este tipo de investigaciones ha cuestionado la validez estricta de la Hipótesis del Mercado Eficiente (HME), formulada por Fama (1970), que sostiene que los precios de los activos financieros reflejan toda la información disponible y que, por tanto, no es posible obtener rendimientos superiores al promedio del mercado mediante el análisis de información pública. Sin embargo, la incorporación de análisis de sentimientos sugiere que las emociones, sesgos cognitivos y percepciones del mercado —extraídas de fuentes textuales— pueden proporcionar señales adicionales para anticipar movimientos de precios, particularmente en horizontes de corto plazo (Barberis, Shleifer & Vishny, 1998; Shiller, 2017).

El desarrollo de modelos avanzados de PLN, como BERT (Bidirectional Encoder Representations from Transformers) y sus variantes especializadas como FinBERT, ha revolucionado el análisis de textos financieros. Estos modelos permiten capturar matices y contextos específicos del lenguaje económico, mejorando la precisión en la clasificación de sentimientos y la predicción de movimientos de mercado. FinBERT, entrenado con grandes volúmenes de textos financieros, se ha consolidado como una herramienta eficaz para extraer señales de mercado desde noticias económicas (Yang et al., 2020).

En cuanto a la arquitectura de los modelos, las redes neuronales recurrentes (RNN) y las redes de memoria a corto y largo plazo (LSTM) han demostrado su utilidad en la predicción de series temporales financieras. Estas redes pueden modelar dependencias temporales complejas, lo cual es crucial para analizar la evolución del sentimiento a lo largo del tiempo y su correlación con los precios de mercado (Fischer & Krauss, 2018). También se han explorado modelos híbridos, como las redes convolucionales (CNN) combinadas con LSTM, para mejorar la extracción de características en textos especializados (Akhtar et al., 2020).

Dentro del contexto colombiano, el COLCAP es el principal índice bursátil del país, compuesto por las acciones más líquidas y representativas de la Bolsa de Valores de Colombia. Este índice actúa como un barómetro del mercado accionario nacional y es utilizado por inversionistas, analistas y entidades financieras para monitorear la salud del mercado. El estudio del sentimiento asociado a las noticias económicas en Colombia puede aportar una dimensión adicional de análisis para interpretar la evolución del COLCAP, especialmente en coyunturas de incertidumbre económica o política.

En Colombia, la literatura académica sobre análisis de sentimientos en mercados financieros es aún incipiente. Algunas investigaciones preliminares han encontrado correlaciones entre el tono de las noticias económicas nacionales y ciertos movimientos en el mercado accionario, pero los estudios se han limitado a métodos tradicionales, sin incorporar arquitecturas modernas de PLN ni aprendizaje automático. Esto representa una brecha de conocimiento y una oportunidad para generar aportes sustanciales al entendimiento del mercado bursátil colombiano a través de técnicas computacionales avanzadas.

En comparación con otros índices latinoamericanos como el BOVESPA (Brasil) o el IPC (México), el COLCAP presenta ciertas características particulares, como una menor liquidez y una concentración en pocas empresas que dominan la capitalización bursátil. Esto sugiere que los flujos de información y el sentimiento generado en los medios pueden tener un efecto más

pronunciado sobre su comportamiento. Además, el consumo de información financiera en Colombia tiende a concentrarse en un número reducido de medios económicos y plataformas digitales, lo que facilita la recopilación de datos pero también puede introducir sesgos informativos si no se considera una muestra representativa.

El crecimiento exponencial de fuentes digitales ha permitido acceder a grandes volúmenes de datos no estructurados como noticias, tweets, blogs financieros y foros especializados. En particular, las noticias formales publicadas por medios económicos han demostrado ser más fiables para construir modelos predictivos, ya que presentan estructuras gramaticales más consistentes y un lenguaje técnico que facilita la extracción semántica (Rao & Srivastava, 2012).

Desde el punto de vista técnico, uno de los mayores retos para aplicar PLN en español es la escasez de corpus etiquetados y diccionarios financieros especializados. A diferencia del inglés, que cuenta con recursos como el Loughran-McDonald Financial Sentiment Dictionary, en español todavía se requieren más iniciativas para desarrollar herramientas de anotación que capturen la polaridad contextual del lenguaje económico. Esta situación limita el rendimiento de modelos generalistas, los cuales no siempre capturan correctamente la semántica financiera.

El aprendizaje automático, como rama de la inteligencia artificial, permite construir modelos capaces de aprender patrones a partir de grandes volúmenes de datos, sin necesidad de programación explícita. En el campo financiero, estos modelos se han utilizado para realizar tareas como la clasificación de noticias, la predicción de precios, la segmentación de clientes o la detección de fraudes. Algoritmos como Support Vector Machines (SVM), XGBoost y Random Forest han mostrado buen desempeño en tareas de clasificación de sentimientos (Huang et al., 2014), particularmente cuando se complementan con representaciones vectoriales del lenguaje como TF-IDF, Word2Vec o embeddings derivados de transformers.

Estudios han demostrado que el análisis de sentimientos puede predecir movimientos del mercado con una precisión significativa. Por ejemplo, Li et al. (2021) encontraron que los sentimientos extraídos de titulares de noticias eran capaces de anticipar movimientos de precios con mayor exactitud que algunos modelos técnicos basados únicamente en series numéricas. Esto confirma que la narrativa mediática no solo refleja el estado del mercado, sino que puede anticipar reacciones futuras.

Entre los principales desafíos del análisis de sentimientos en finanzas se encuentran: la ambigüedad del lenguaje, la ironía o el sarcasmo, la necesidad de modelos adaptativos para diferentes contextos, y los problemas éticos asociados al uso de IA en decisiones económicas. Existen preocupaciones válidas sobre la posibilidad de manipular el mercado mediante la difusión estratégica de noticias sesgadas o alarmistas que impacten modelos automatizados. Además, la falta de transparencia en el funcionamiento de algunos algoritmos puede dificultar la trazabilidad y validación de sus predicciones.

Finalmente, aunque existen avances relevantes en mercados desarrollados, aún es escasa la literatura empírica centrada en el análisis de sentimiento aplicado al COLCAP. Por tanto, la implementación de una metodología que combine PLN, aprendizaje automático y fuentes textuales nacionales representa una contribución innovadora. Este enfoque no solo busca establecer una correlación entre el sentimiento del mercado y las variaciones del índice, sino también proporcionar una herramienta replicable que pueda ser utilizada por analistas, inversionistas institucionales y entidades académicas interesadas en el comportamiento de los mercados emergentes.

Adicionalmente, el análisis de sentimientos puede desempeñar un papel crucial en la gestión de portafolios, especialmente en escenarios de alta volatilidad. Durante periodos de incertidumbre, como crisis políticas, cambios regulatorios o fenómenos macroeconómicos inesperados, el comportamiento de los inversionistas suele estar fuertemente influenciado por

la percepción emocional del entorno. En este contexto, contar con herramientas que permitan anticipar reacciones del mercado a partir del lenguaje mediático puede mejorar la capacidad de los gestores para tomar decisiones informadas y oportunas, ajustando su exposición al riesgo según las señales extraídas del entorno informativo.

Asimismo, los reguladores financieros y organismos supervisores podrían beneficiarse de este tipo de metodologías para monitorear en tiempo real el clima emocional de los mercados. Esto permitiría una respuesta más ágil ante eventos de pánico financiero o euforia especulativa, contribuyendo a la estabilidad del sistema financiero. De igual forma, en el ámbito de la inversión socialmente responsable (ISR), el análisis de sentimientos puede ayudar a evaluar la percepción pública sobre el comportamiento ético y sostenible de las empresas, alineando decisiones de inversión con criterios ambientales, sociales y de gobernanza (ESG). Por tanto, esta línea de investigación ofrece un alto potencial de impacto tanto académico como práctico en economías emergentes como la colombiana.

Metodología

Primer Nivel

Enfoque, alcance y diseño de la investigación

Para el desarrollo del presente trabajo, se adopta un enfoque cuantitativo, ya que se parte del análisis de datos estructurados (índice bursátil) y no estructurados (texto de noticias) con el fin de medir variables y establecer relaciones estadísticas entre ellas. En este enfoque se aplican técnicas computacionales, como el Procesamiento de Lenguaje Natural (PLN) y modelos de aprendizaje automático, para clasificar el sentimiento y correlacionarlo con el comportamiento de los mercados financieros.

El alcance de la investigación es correlacional y aplicado ya que de un lado busca identificar y cuantificar la relación existente entre dos variables: el sentimiento de las noticias y las variaciones del índice COLCAP y por otra parte, se busca desarrollar un modelo replicable que pueda ser utilizado como herramienta de análisis para inversionistas, analistas financieros o instituciones del mercado de valores.

El diseño de investigación es no experimental ya que no hay manipulación deliberada de variables independientes, sino que se observan en su contexto natural, es transversal porque la recolección de datos se realiza en un único periodo de análisis retrospectivo (noticias y valores históricos del COLCAP entre 2022 y 2024) y, es también correlacional porque se pretende analizar cómo se relacionan estadísticamente las variables sin establecer causalidad directa.

Definición de variables

Variable	Definición conceptual	Definición operacional	Dimensiones
Sentimiento en noticias financieras	Es la actitud emocional (positiva, negativa o neutral) expresada en noticias periodísticas con enfoque financiero. Se deriva del enfoque del análisis de sentimientos dentro del PLN, el cual permite detectar subjetividad en el lenguaje (Liu, 2012).	Se mide mediante técnicas de Procesamiento de Lenguaje Natural, utilizando modelos como FinBERT para clasificar titulares y cuerpos de noticias. Los resultados se codifican en valores de polaridad (positivo, negativo o neutral).	Polaridad (positiva, negativa, neutral); Tendencia temporal; Frecuencia
Índice COLCAP	Es el principal índice accionario del mercado bursátil colombiano, compuesto por las acciones más líquidas de la Bolsa de Valores de Colombia (BVC). Representa el comportamiento agregado del mercado.	Se mide mediante la variación porcentual diaria del COLCAP, tomando como fuente los datos históricos publicados por la BVC. Se utilizará como variable numérica continua.	Rendimiento (%); Volatilidad; Tendencia mensual
Relación sentimiento–COLCAP	Grado de asociación entre el sentimiento expresado en noticias y el comportamiento bursátil reflejado en el COLCAP. Se entiende desde una perspectiva de interdependencia entre percepción del mercado e índices financieros.	Se mide a través de coeficientes de correlación (como Pearson o Spearman) entre el puntaje de sentimiento y las variaciones del índice COLCAP, evaluadas en ventanas temporales similares.	Correlación; Coincidencia temporal; Magnitud de efecto

Población y muestra

La población objeto de estudio en esta investigación está compuesta por todas las noticias financieras publicadas entre los años 2022 y 2024 en medios de comunicación económicos relevantes, principalmente colombianos que aborden temas relacionados con el mercado bursátil nacional y, en particular, con el comportamiento de las empresas listadas en

el índice COLCAP. Esta población también incluye los registros históricos diarios del índice COLCAP publicados por la Bolsa de Valores de Colombia (BVC) durante el mismo periodo.

Dado el amplio volumen de noticias publicadas en medios digitales a lo largo de tres años, y la naturaleza digital del análisis, se ha optado por un muestreo no probabilístico por conveniencia, seleccionando aquellas noticias que incluyen palabras clave relacionadas con el COLCAP, acciones colombianas, análisis bursátil o empresas que lo componen. Estas noticias serán recolectadas mediante técnicas de scraping automatizado desde fuentes como Portafolio, La República, Valora Analitik, El Tiempo, Dinero, Investing y Semana.

La muestra estará compuesta por aproximadamente 1.000 noticias clasificadas mediante procesamiento de lenguaje natural, las cuales serán cruzadas con las variaciones del índice COLCAP en las fechas correspondientes. En el caso de la serie de datos financieros, se tomará la totalidad de los registros diarios del COLCAP desde enero de 2022 hasta diciembre de 2024.

Este enfoque permite establecer una relación temporal entre el sentimiento del mercado expresado en noticias y el comportamiento real del índice, garantizando una base de análisis representativa y adecuada para un estudio correlacional.

Segundo nivel

Selección de métodos o instrumentos para recolección de información

En el presente estudio correlacional, se utilizarán instrumentos tecnológicos y computacionales validados en investigaciones previas para la recolección y medición de las variables definidas: sentimiento en noticias financieras y comportamiento del índice COLCAP. El objetivo es garantizar consistencia, precisión y pertinencia en la medición, acorde con el enfoque cuantitativo y la naturaleza no experimental de la investigación.

Para la recolección de información textual, se utilizará un scraper automatizado desarrollado en Python, el cual recopilará noticias financieras publicadas entre 2022 y 2024 desde medios nacionales reconocidos como Portafolio, La República, Dinero, Valora Analitik, Semana, El Tiempo, Investing y El Espectador. Este instrumento aplicará filtros por palabra clave para identificar textos relevantes relacionados con el índice COLCAP. El script de scraping será diseñado por el grupo de investigación y se anexará al final del documento como parte del instrumento de recolección.

Una vez recolectadas las noticias, se aplicará el modelo FinBERT, un instrumento ampliamente validado para el análisis de sentimiento en textos financieros. FinBERT ha sido entrenado específicamente en datos económicos y ha demostrado resultados sólidos en la clasificación de polaridad textual (positivo, negativo o neutral). Este modelo permite asignar una etiqueta de sentimiento a cada noticia, la cual será luego vinculada con la fecha correspondiente del comportamiento del índice COLCAP.

En cuanto a la recolección de los datos financieros, se utilizarán los registros históricos diarios del índice COLCAP publicados oficialmente por la Bolsa de Valores de Colombia (BVC), disponibles en su plataforma web. Esta fuente de información constituye un instrumento confiable y estandarizado para medir el comportamiento del mercado colombiano.

Por tanto, los instrumentos seleccionados —scraper de noticias, modelo FinBERT para clasificación de sentimiento y serie histórica del COLCAP—los cuales, además, permiten una medición cuantitativa replicable, suficiente y válida para establecer relaciones entre percepción de los usuarios del mercado y el desempeño bursátil. Los códigos y procedimientos utilizados se anexarán al final del trabajo como parte de los instrumentos metodológicos.

Técnicas de análisis de datos

En correspondencia con los objetivos planteados y las variables definidas en el estudio, las técnicas seleccionadas deben garantizar la validez y confiabilidad de los resultados, así como su pertinencia respecto al enfoque y diseño de investigación adoptados. Por lo tanto, en este estudio, de tipo cuantitativo, no experimental, transversal y correlacional, el análisis de datos se orienta a explorar la relación entre el sentimiento expresado en noticias financieras y el comportamiento del índice bursátil COLCAP en Colombia durante el periodo 2022–2024. Para tal fin, se emplean técnicas de procesamiento de texto, análisis estadístico descriptivo, análisis de series temporales y métodos de correlación. Los cuales se desarrollarán de la siguiente forma:

1. Procesamiento de texto y análisis de sentimiento

La primera etapa del análisis consiste en transformar el corpus de noticias financieras recolectadas mediante scraping en una base de datos estructurada que permita aplicar modelos de Procesamiento de Lenguaje Natural (PLN). Esta etapa del análisis se trata de un preprocesamiento textual, y comprende las siguientes operaciones: limpieza de texto (eliminación de signos de puntuación, URLs, caracteres especiales), normalización (minúsculas, eliminación de stopwords), tokenización (división del texto en palabras o frases), y vectorización (transformación del texto en representaciones numéricas).

Posteriormente, se aplica el modelo FinBERT, una arquitectura basada en Transformers entrenada específicamente para análisis de sentimiento en textos financieros. FinBERT es un modelo que se ajusta bastante al enfoque de este análisis ya que permite clasificar cada noticia como positiva, negativa o neutral, asignando una probabilidad a cada clase. Esta clasificación se almacena junto con la fecha de la noticia y otros metadatos como el medio de origen o el titular.

El análisis de sentimiento se expresa finalmente en términos numéricos mediante una codificación ordinal: positivo (1), neutral (0), negativo (–1). Esto permite cuantificar el sentimiento predominante en el periodo a analizar y calcular medidas agregadas como frecuencia, proporción y promedio de sentimiento.

2. Análisis estadístico descriptivo

Una vez obtenidos los datos de sentimiento codificados, se procede a aplicar técnicas de estadística descriptiva tanto para las variables textuales (ya transformadas) como para la variable numérica continúa representada por el índice COLCAP. Las técnicas utilizadas incluyen:

- Frecuencia absoluta y relativa de cada tipo de sentimiento.
- Medidas de tendencia central (media, mediana y moda) para la polaridad del sentimiento agregado.
- Medidas de dispersión (desviación estándar, varianza) para identificar la volatilidad del sentimiento.
- Visualizaciones como histogramas, gráficos de líneas y diagramas de cajas para observar la distribución temporal del sentimiento y del COLCAP.

La idea principal en esta parte del análisis preliminar es caracterizar la muestra, identificar patrones iniciales y detectar posibles outliers o inconsistencias antes del análisis correlacional.

3. Análisis de series temporales

Tanto el sentimiento como el valor del índice COLCAP tienen una estructura temporal, por lo que su análisis requiere técnicas que consideren el orden cronológico y la dinámica evolutiva de las observaciones. Para ello, se emplean herramientas básicas de análisis de series temporales, como:

- Suavizamiento exponencial o medias móviles para observar tendencias generales.
- Descomposición de series en componentes de tendencia, estacionalidad y ruido.
- Correlación cruzada (cross-correlation) para analizar posibles desfases entre el sentimiento y las variaciones del COLCAP.

Estas técnicas permiten evaluar si el sentimiento en las noticias anticipa, acompaña o sigue los movimientos del índice bursátil, lo cual es clave para responder a la pregunta de investigación.

4. Análisis de correlación

El objetivo central de esta investigación es determinar la existencia y naturaleza de una relación entre dos variables: el sentimiento expresado en noticias financieras y el comportamiento del índice COLCAP. Para ello se emplean técnicas de análisis de correlación, específicamente las siguientes:

- Coeficiente de correlación de Pearson, cuando las variables cumplen los supuestos de normalidad y linealidad. Este coeficiente mide la fuerza y dirección de la relación lineal entre dos variables continuas.
- Coeficiente de correlación de Spearman, en caso de que alguna de las variables no cumpla con los criterios paramétricos, o si la relación es monótonamente creciente o decreciente pero no lineal.
- Ambos coeficientes producen valores entre –1 y 1, donde valores cercanos a ±1 indican relaciones fuertes, y valores cercanos a 0 indican ausencia de relación. Se evaluarán también los valores de p (significancia estadística) para determinar si la correlación observada puede atribuirse al azar.

5. Validación de resultados y robustez

Con el fin de asegurar la robustez del análisis, se realizarán pruebas complementarias como:

- Análisis de sensibilidad al cambiar el modelo de análisis de sentimiento (por ejemplo, comparar FinBERT con otro modelo como TextBlob adaptado al dominio financiero).
- Revisión de casos discordantes, es decir, días con sentimiento muy positivo y caída del índice, o viceversa, para explorar explicaciones contextuales.

6. Herramientas y software utilizados

Las técnicas descritas serán implementadas utilizando software especializado, entre ellos:

- Python: para scraping, procesamiento de texto, clasificación con FinBERT y análisis estadístico (librerías como pandas, scikit-learn, statsmodels, matplotlib).
- Excel y Power BI: para validación visual y elaboración de gráficas interactivas de resultados.
- Word: Como entorno de documentación reproducible y resumen del análisis.

Informe Técnico final de investigación

Luego de realizar el procesamiento y limpieza de datos y de aplicar los modelos de de procesamiento de Lenguaje Natural (NLP) en el que se busco determinar el sentimiento de cada noticia recolectada, se obtuvieron los siguientes resultados:

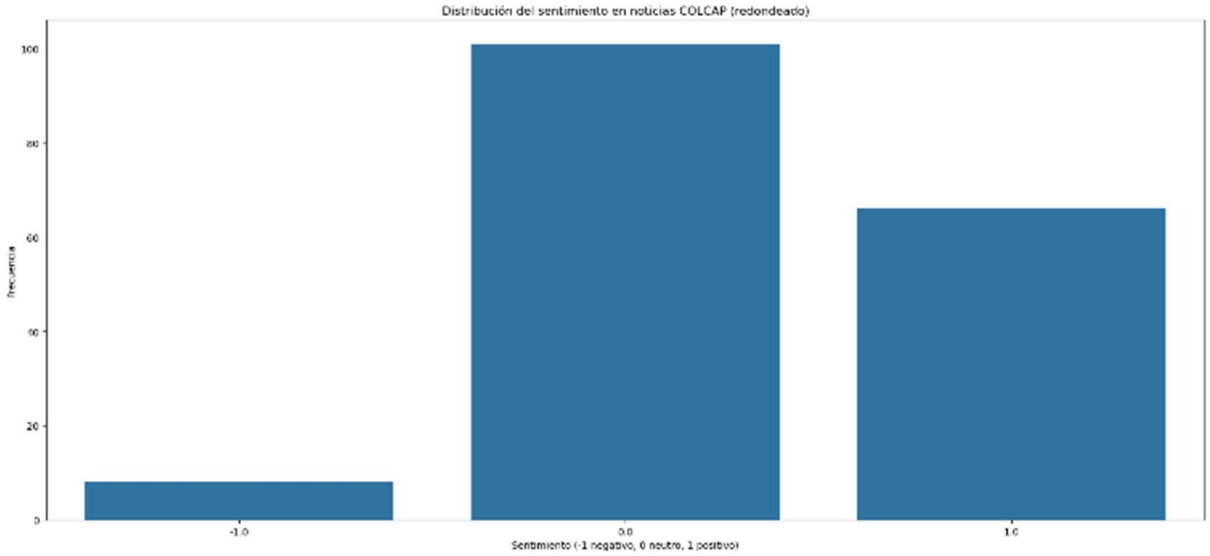

Figura 1: Distribución del sentimiento en noticias COLCAP

Como se observa en la visualización generada, para el periodo analizado el sentimiento asociado a las noticias financieras no se comporta como una variable significativa para explicar las variaciones del índice de COLCAP en la bolsa de valores de Colombia. Para confirmar esta observación, se realizó un análisis de correlación entre la clasificación del sentimiento y la variación diaria porcentual del índice COLCAP.

Resultado del análisis de correlación:

- Coeficiente de correlación de Pearson: 0.0487
- Valor -p: 0.5218

Dentro de los resultados del coeficiente de correlación indica una relación positiva extremadamente débil lo que sugiere que existe una ausencia de relación lineal entre el sentimiento de la noticia y la variación del COLCAP, por otra parte el valor p indica que la correlación no es estadísticamente significativa y por lo tanto no hay evidencia suficiente para afirmar que existe una relación lineal entre las dos variables.

Posteriormente, se procede a realizar una regresión lineal entre la clasificación del sentimiento y la variación porcentual del COLCAP obteniento lo siguiente:

R^2 = 0.0024

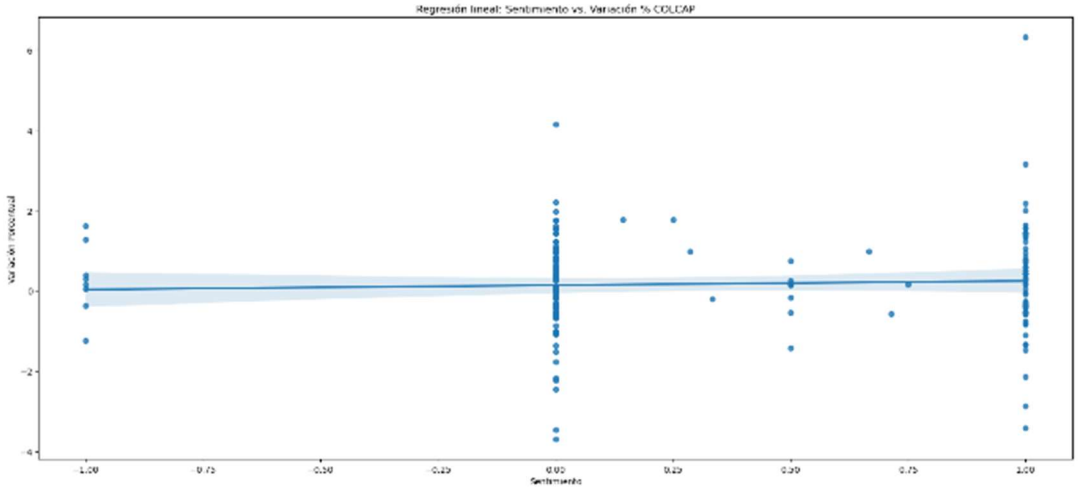

Figura 2: Regresion Lineal: Sentimiento vs variación % COLCAP

Lo cual indica que solo el 0.24% de la variabilidad en la variación porcentual del COLCAP es explicada por el sentimiento de las noticias lo que sugiere que existen otras variables mucho más importantes que influyen en el comportamiento del COLCAP.

Estos resultados nos permiten dar respuesta a la pregunta inicialmente propuesta para este trabajo de investigación, indicando que en los días seleccionados de los últimos 3 años, la tendencia princial del sentimiento en las noticias financieras es neutro pese a los movimientos económicos y las fluctuaciones y que estos presentan poca relación con los movimientos del COLCAP en la bolsa de valores.

Discusión

El presente estudio tuvo como objetivo principal analizar la tendencia del sentimiento en noticias financieras y su relación con las variaciones del índice COLCAP en los últimos tres años. Para ello, se implementaron técnicas de Procesamiento de Lenguaje Natural (NLP) y modelos predictivos, permitiendo cuantificar y modelar dicha relación. Esta sección discute los principales hallazgos obtenidos, interpretándolos a la luz del marco teórico previamente establecido, y reflexiona sobre la aplicabilidad y limitaciones de las propuestas metodológicas empleadas.

Los datos obtenidos revelan que la mayoría de las noticias financieras presentan una polaridad neutra o levemente positiva. Esta tendencia coincide con lo planteado por Tetlock (2007), quien argumenta que los medios financieros tienden a mantener un tono informativo conservador, salvo en contextos de crisis o incertidumbre. La clasificación de sentimiento fue realizada con herramientas como TextBlob y VADER, que si bien ofrecen un enfoque léxico simple, fueron adecuados para una primera aproximación.
El análisis de correlación mostró una débil relación lineal entre el sentimiento promedio de las noticias y la variación porcentual diaria del índice COLCAP, bajo estos hechos, se pudo argumentar que la importancia de las variables demuestra que el sentimiento

tiene un peso muy bajo en la predicción del índice. Así, el sentimiento agregado no influye significativamente en el corto plazo sobre el índice, especialmente en mercados emergentes donde la información puede no ser procesada instantáneamente por todos los agentes.

De acuerdo a los hallazgos encontrados, y como propuestas de intervención y modelos de aplicación se tiene que si bien el sentimiento no resultó ser un predictor fuerte por sí solo, su utilidad puede verse ampliada en modelos híbridos que incorporen variables económicas fundamentales, datos técnicos y eventos exógenos (como reformas fiscales o coyunturas políticas). Además, una posible intervención o una aplicación para un estudio posterior sería el desarrollo de un sistema de alerta temprana basado en picos de sentimiento negativo, que podría anticipar días de alta volatilidad, lo cual se plantea en diversos estudios como el planteado por Shiller (2005).

Finalmente, Es importante reconocer que el corpus de noticias recolectado tiene limitaciones en cobertura y que la frecuencia de noticias al ser aleatoria fue irregular, lo que puede haber afectado la robustez estadística en ciertos períodos. Se recomienda en futuros trabajos incorporar fuentes adicionales (redes sociales, informes técnicos, discursos económicos) para mejorar la representatividad. Además, se puede concluir que en conexión con la teoría previamente estudiada, los resultados sugieren que, en línea con la teoría conductual (Barberis & Thaler, 2003), el sentimiento puede tener un papel más relevante en contextos de pánico o euforia, donde el comportamiento irracional de los inversionistas toma protagonismo. Finalmente, esta investigación abre una puerta hacia la integración de análisis de texto en la economía financiera colombiana, proponiendo un enfoque novedoso y complementario a los modelos tradicionales de análisis técnico y fundamental que se han trabajado previamente.

Referencias

- Bollen, J., Mao, H., & Zeng, X. (2011). Twitter mood predicts the stock market. Journal of Computational Science, 2(1), 1–8. https://doi.org/10.1016/j.jocs.2010.12.007
- Shynkevich, Y., McGinnity, T. M., Coleman, S. A., & Belatreche, A. (2016). Forecasting price movements using technical indicators: Investigating the impact of varying input window length. Decision Support Systems, 89, 102–114. https://doi.org/10.1016/j.dss.2016.03.001
- Mao, H., Counts, S., & Bollen, J. (2011). Predicting financial markets: Comparing survey, news, Twitter and search engine data. arXiv preprint arXiv:1112.1051.
- Bollen, J., Mao, H., & Zeng, X. (2011). Twitter mood predicts the stock market. Journal of Computational Science, 2(1), 1–8. https://doi.org/10.1016/j.jocs.2010.12.007
- Shynkevich, Y., McGinnity, T. M., Coleman, S. A., & Belatreche, A. (2016). Predicting stock price movements based on different categories of news articles using machine learning classifiers. Neurocomputing, 272, 578–588.
- Tetlock, P. C. (2007). Giving content to investor sentiment: The role of media in the stock market. The Journal of Finance, 62(3), 1139–1168. https://doi.org/10.1111/j.1540-6261.2007.01232.x
- Loughran, T., & McDonald, B. (2011). When is a liability not a liability? Textual analysis, dictionaries, and 10-Ks. The Journal of Finance, 66(1), 35–65. https://doi.org/10.1111/j.1540-6261.2010.01625.x
- García, D. (2013). Sentiment during recessions. The Journal of Finance, 68(3), 1267–1300.
- Barberis, N., Shleifer, A., & Vishny, R. (1998). A model of investor sentiment. Journal of Financial Economics, 49(3), 307–343. https://doi.org/10.1016/S0304-405X(98)00027-0

- Kahneman, D., & Tversky, A. (1979). Prospect theory: An analysis of decision under risk. Econometrica, 47(2), 263–291. https://doi.org/10.2307/1914185
- Shiller, R. J. (2017). Narrative economics. American Economic Review, 107(4), 967–1004. https://doi.org/10.1257/aer.107.4.967
- Akhtar, M. S., Ekbal, A., & Cambria, E. (2020). How intense are you? Predicting intensities of emotions using CNN and LSTM. Knowledge-Based Systems, 222, 106994.
- Araci, D. (2019). FinBERT: Financial sentiment analysis with pre-trained language models. arXiv preprint arXiv:1908.10063.
- Barberis, N., Shleifer, A., & Vishny, R. (1998). A model of investor sentiment. Journal of Financial Economics, 49(3), 307–343.
- Bollen, J., Mao, H., & Zeng, X. (2011). Twitter mood predicts the stock market. Journal of Computational Science, 2(1), 1–8.
- Chen, C., et al. (2020). Sentiment analysis for financial texts. Information Processing & Management, 57(1), 102117.
- Fama, E. F. (1970). Efficient capital markets: A review of theory and empirical work. The Journal of Finance, 25(2), 383–417.
- Fischer, T., & Krauss, C. (2018). Deep learning with long short-term memory networks for financial market predictions. European Journal of Operational Research, 270(2), 654–669.
- Hafez, Y., et al. (2023). Sentiment analysis in stock market research: A bibliometric study. Information, 14(2), 106.
- Huang, W., Nakamori, Y., & Wang, S. Y. (2014). Forecasting stock market movement direction with support vector machine. Computers & Operations Research, 32(10), 2513–2522.

- Kordonis, I., Symeonidis, A. L., & Arampatzis, A. (2016). Stock movement prediction with news sentiment analysis. ACM Transactions on Information Systems (TOIS), 35(2), 1–25.
- Li, X., Xie, H., Chen, L., Wang, J., & Deng, X. (2021). News impact on stock price return via sentiment analysis. Knowledge-Based Systems, 196, 105824.
- Rao, T., & Srivastava, S. (2012). Analyzing stock market movements using Twitter sentiment analysis. Proceedings of the 2012 IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology, 3, 99–103.
- Shiller, R. J. (2017). Narrative economics. American Economic Review, 107(4), 967–1004.
- Sun, B., et al. (2022). A review of deep learning-based sentiment analysis in stock market. Engineering Applications of Artificial Intelligence, 109, 104653.
- Tetlock, P. C. (2007). Giving content to investor sentiment: The role of media in the stock market. The Journal of Finance, 62(3), 1139–1168.
- Yang, X., & Wang, J. (2020). Financial sentiment analysis based on deep learning: A comparative study. IEEE Access, 8, 138254–138264.
- Zhang, Y., Fuehres, H., & Gloor, P. (2018). Predicting stock market indicators through Twitter “I hope it is not as bad as I fear”. Procedia - Social and Behavioral Sciences, 26, 55–62.

Anexos

1. **Scraper de Python para encontrar noticias relevantes:**

```
"""
Scraper para encongtrar notivcias financieras relacionadas con el COLCAP (2022-2024)
para análisis de sentimientos.
"""

import requests
from bs4 import BeautifulSoup
import pandas as pd
import time
import os
import re
import random
from concurrent.futures import ThreadPoolExecutor, as_completed
from datetime import datetime
from dateutil import parser
from cachetools import TTLCache
from tqdm import tqdm
import logging
from selenium import webdriver
from selenium.webdriver.chrome.service import Service
from selenium.webdriver.chrome.options import Options
try:
    import chromedriver_autoinstaller
except ImportError:
    chromedriver_autoinstaller = None
import locale

# Configurar locale para español
try:
    locale.setlocale(locale.LC_TIME, 'es_ES.UTF-8')
except locale.Error:
    try:
        locale.setlocale(locale.LC_TIME, 'Spanish_Spain.1252')
    except locale.Error:
        print("No se pudo configurar el locale para español. Puede afectar el parseo de
fechas.")

# ----------------------------------------
# CONFIGURACIÓN
# ----------------------------------------
USER_AGENTS = [
    "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like
Gecko) Chrome/91.0.4472.124 Safari/537.36",
    "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/537.36 (KHTML, like
Gecko) Chrome/91.0.4472.114 Safari/537.36",
```

```
    "Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:89.0) Gecko/20100101 Firefox/89.0"
]

KEYWORDS = [
    "colcap", "bolsa de valores", "mercado bursátil", "acciones colombianas", "índice
accionario",
    "bvc", "bolsa colombiana","grupo argos", "ecopetrol", "bancolombia", "grupo sura",
"isa", "nutresa",
    "cemargos", "avianca", "grupo exito", "terpel", "canacol", "cemex", "etb", "banco de
bogotá", "davivienda", "grupo aval",
    "grupo bolivar", "corficolombiana", "grupo energía bogotá", "celsia", "promigas",
"mineros", "pei", "sube colcap", "baja colcap",
    "caída", "repunte", "alza", "tendencia", "comportamiento", "fluctuación", "reacción del
mercado", "bursátil", "cierre de mercado",
    "apertura de mercado", "rendimiento", "uplift", "ROI", "inversión", "superfinanciera",
"superintendencia financidera", "corredores de bolsa",
    "zona de soporte", "rebalanceo", "mercado accionario", "renta variable", "inversión",
"finanzas", "economía colombiana",
    "análisis técnico", "análisis fundamental", "volatilidad", "sentimiento del mercado",
    "indicadores económicos", "informe financiero", "resultados corporativos", "elecciones
presidenciales"
]
KEYWORDS = [kw.lower() for kw in KEYWORDS]
SPANISH_WORDS = set(["el", "la", "los", "las", "de", "en", "y", "a", "que", "con", "para",
"es"])

BUSQUEDAS = KEYWORDS

OUTPUT_PATH = r"C:\Users\luisa\OneDrive - universidadean.edu.co\Trabajo final
especialización\Scraping\Noticias_COLCAP_2022_2024.xlsx"
LOG_PATH = r"C:\Users\luisa\OneDrive - universidadean.edu.co\Trabajo final
especialización\Scraping\log_scraper_colcap.txt"
NEWSAPI_KEY = "TU_API_KEY_AQUI"  # Reemplaza con tu clave válida de NewsAPI

# Configurar ChromeDriver
if chromedriver_autoinstaller:
    chromedriver_autoinstaller.install()
chrome_options = Options()
chrome_options.add_argument("--headless")
chrome_options.add_argument("--disable-gpu")
chrome_options.add_argument(f"user-agent={random.choice(USER_AGENTS)}")

# Iniciar cache and logging
cache = TTLCache(maxsize=1000, ttl=3600)
logging.basicConfig(filename=LOG_PATH, level=logging.INFO, format='%(asctime)s -
%(levelname)s - %(message)s')

# ----------------------------------------
# FUNCIONES AUXILIARES
```

```
# ----------------------------------------
def contiene_palabra_clave(texto):
    if not texto:
        return False
    texto = texto.lower()
    return any(kw in texto for kw in KEYWORDS)

def limpiar_texto(texto):
    return re.sub(r'\s+', ' ', texto).strip() if texto else ""

def es_probablemente_espanol(texto):
    """Heurística simple para detectar si el texto es en español."""
    if not texto:
        return False
    palabras = set(re.sub(r'[^\w\s]', '', texto.lower()).split())
    return len(palabras & SPANISH_WORDS) / len(palabras) > 0.2  # Si más del 20% son
palabras en esp

def extraer_fecha(soup, link=""):
    """
    Extract publication date from HTML document, only for January 2022.
    """
    date_selectors = [
        'time',
        'meta[property="article:published_time"]',
        'span.date',
        'div.date',
        'div.datetime',
        'span.published-date',
        'span[itemprop="datePublished"]'
    ]
    for tag in date_selectors:
        elements = soup.select(tag)
        for elem in elements:
            text = elem.get_text(strip=True) or elem.get('content', '') or elem.get('datetime', '')
            if not text:
                continue
            text = re.sub(r'\d{1,2}:\d{2}\s*(a\.m\.|p\.m\.)', '', text,
flags=re.IGNORECASE).strip()
            logging.info(f"Texto de fecha extraído en {link}: '{text}'")
            try:
                fecha = parser.parse(text, fuzzy=True, dayfirst=True)
                # Filtrar solo para enero de 2022
                if fecha.year == 2022 and fecha.month == 1:
                    logging.info(f"Fecha parseada en {link}: {fecha.strftime('%Y-%m-%d')}")
                    return fecha.strftime('%Y-%m-%d')
                else:
                    logging.warning(f"Fecha fuera de enero 2022: {text} (parseada como
{fecha}) en {link}")
```

```
            except (ValueError, TypeError) as e:
                logging.warning(f"Error al parsear fecha: {text} en {link} - Error: {e}")
        logging.warning(f"No se encontró fecha válida en {link} para enero 2022")
        return "No disponible"

def registrar_log(fuente, total, errores, inicio):
    fin = datetime.now().strftime("%Y-%m-%d %H:%M:%S")
    with open(LOG_PATH, "a", encoding="utf-8") as f:
        f.write(f"Fuente: {fuente}\nInicio: {inicio}\nFin: {fin}\nNoticias filtradas:
{total}\nErrores: {errores}\n\n")
    logging.info(f"{fuente} - Noticias: {total}, Errores: {errores}")

# ----------------------------------------
# SCRAPER GENÉRICO
# ----------------------------------------
def scrape_generico(nombre_fuente, url_busqueda, dominio, selector_items,
use_selenium=False):
    noticias = []
    errores = 0
    inicio = datetime.now().strftime("%Y-%m-%d %H:%M:%S")
    visited_urls = set()

    def scrape_pagina(keyword, page):
        nonlocal errores
        local_noticias = []
        url = url_busqueda.format(keyword=keyword, page=page)
        try:
            if url in cache:
                return cache[url]
            headers = {"User-Agent": random.choice(USER_AGENTS)}
            print(f"Buscando en: {url}")
            if use_selenium:
                try:
                    if 'CHROMEDRIVER_PATH' in globals():
                        service = Service(CHROMEDRIVER_PATH)
                        driver = webdriver.Chrome(service=service, options=chrome_options)
                    else:
                        driver = webdriver.Chrome(options=chrome_options)
                    driver.get(url)
                    time.sleep(random.uniform(2, 4))
                    soup = BeautifulSoup(driver.page_source, 'html.parser')
                    driver.quit()
                except Exception as e:
                    errores += 1
                    logging.error(f"Error Selenium en {url}: {e}")
                    return local_noticias
            else:
                r = requests.get(url, headers=headers, timeout=10)
                print(f"Estado: {r.status_code}")
```

```
            r.raise_for_status()
            soup = BeautifulSoup(r.content, 'html.parser')
        items = soup.select(selector_items)
        print(f"Elementos encontrados en {url}: {len(items)}")
        for tag in items:
            titulo = limpiar_texto(tag.get_text(strip=True))
            if not contiene_palabra_clave(titulo):
                continue
            link = tag.get("href", "")
            if not link.startswith("http"):
                link = dominio + link
            if link in visited_urls or link in cache:
                continue
            visited_urls.add(link)
            try:
                if use_selenium:
                    if 'CHROMEDRIVER_PATH' in globals():
                        service = Service(CHROMEDRIVER_PATH)
                        driver = webdriver.Chrome(service=service, options=chrome_options)
                    else:
                        driver = webdriver.Chrome(options=chrome_options)
                    driver.get(link)
                    time.sleep(random.uniform(2, 4))
                    soup_nota = BeautifulSoup(driver.page_source, 'html.parser')
                    driver.quit()
                else:
                    headers = {"User-Agent": random.choice(USER_AGENTS)}
                    r_nota = requests.get(link, headers=headers, timeout=10)
                    r_nota.raise_for_status()
                    soup_nota = BeautifulSoup(r_nota.content, 'html.parser')
                contenido = limpiar_texto(" ".join(p.get_text(strip=True) for p in
soup_nota.find_all("p")))
                if not contiene_palabra_clave(contenido):
                    continue
                fecha = extraer_fecha(soup_nota, link)
                if fecha == "No disponible":
                    continue
                article = {
                    "titulo": titulo,
                    "url": link,
                    "fuente": dominio,
                    "contenido_articulo": contenido,
                    "fecha_publicacion": fecha
                }
                local_noticias.append(article)
                cache[link] = article
            except (requests.RequestException, Exception) as e:
                errores += 1
                logging.error(f"Error en {link}: {e}")
```

```
                time.sleep(random.uniform(1, 3))
            cache[url] = local_noticias
        except (requests.RequestException, Exception) as e:
            errores += 1
            logging.error(f"Error en {url}: {e}")
        return local_noticias

    with ThreadPoolExecutor(max_workers=3) as executor:
        futures = []
        for keyword in BUSQUEDAS:
            for page in range(1, 4):
                futures.append(executor.submit(scrape_pagina, keyword, page))
        for future in tqdm(as_completed(futures), total=len(futures), desc=f"Scraping
{nombre_fuente}"):
            result = future.result()
            if result:
                noticias.extend(result)

    print(f"{nombre_fuente} → Total noticias filtradas: {len(noticias)}")
    registrar_log(nombre_fuente, len(noticias), errores, inicio)
    return noticias

# ----------------------------------------
# SCRAPER PARA X (TWITTER)
# ----------------------------------------
def scrape_x():
    nombre_fuente = "X (Twitter)"
    noticias = []
    errores = 0
    inicio = datetime.now().strftime("%Y-%m-%d %H:%M:%S")
    visited_urls = set()

    def scrape_x_posts(keyword, page):
        nonlocal errores
        local_noticias = []
        # Usamos la búsqueda de X con fechas específicas para enero 2022
        url = f"https://x.com/search?q={keyword}%20since%3A2022-01-
01%20until%3A2022-01-31&f=live&page={page}"
        try:
            if url in cache:
                return cache[url]
            headers = {"User-Agent": random.choice(USER_AGENTS)}
            print(f"Buscando en: {url}")
            try:
                if 'CHROMEDRIVER_PATH' in globals():
                    service = Service(CHROMEDRIVER_PATH)
                    driver = webdriver.Chrome(service=service, options=chrome_options)
                else:
                    driver = webdriver.Chrome(options=chrome_options)
```

```
                driver.get(url)
                time.sleep(random.uniform(3, 5))
                soup = BeautifulSoup(driver.page_source, 'html.parser')
                driver.quit()
            except Exception as e:
                errores += 1
                logging.error(f"Error Selenium en {url}: {e}")
                return local_noticias

            # Buscar publicaciones (tweets)
            items = soup.select('article div[lang]')
            print(f"Publicaciones encontradas en {url}: {len(items)}")
            for tag in items:
                contenido = limpiar_texto(tag.get_text(strip=True))
                if not contiene_palabra_clave(contenido):
                    continue
                if not es_probablemente_espanol(contenido):
                    logging.info(f"Publicación descartada por no ser probablemente en
español")
                    continue
                # Extraer enlace al tweet
                link_tag = tag.find_parent('a', href=True)
                link = link_tag['href'] if link_tag else ""
                if not link.startswith("http"):
                    link = "https://x.com" + link
                if link in visited_urls or link in cache:
                    continue
                visited_urls.add(link)
                try:
                    # Intentar extraer la fecha del tweet
                    time_tag = tag.find_parent('article').find('time')
                    fecha_text = time_tag['datetime'] if time_tag else ""
                    if fecha_text:
                        fecha = parser.parse(fecha_text).strftime('%Y-%m-%d')
                        # Asegurarse de que está en enero 2022
                        fecha_dt = datetime.strptime(fecha, '%Y-%m-%d')
                        if fecha_dt.year != 2022 or fecha_dt.month != 1:
                            continue
                    else:
                        continue
                    article = {
                        "titulo": contenido[:50] + "..." if len(contenido) > 50 else contenido,  #
Título corto para tweets
                        "url": link,
                        "fuente": "https://x.com",
                        "contenido_articulo": contenido,
                        "fecha_publicacion": fecha
                    }
                    local_noticias.append(article)
```

```
                cache[link] = article
            except (Exception) as e:
                errores += 1
                logging.error(f"Error procesando publicación en {link}: {e}")
            time.sleep(random.uniform(1, 3))
        cache[url] = local_noticias
    except Exception as e:
        errores += 1
        logging.error(f"Error en {url}: {e}")
    return local_noticias

with ThreadPoolExecutor(max_workers=3) as executor:
    futures = []
    for keyword in BUSQUEDAS:
        for page in range(1, 4):
            futures.append(executor.submit(scrape_x_posts, keyword, page))
    for future in tqdm(as_completed(futures), total=len(futures), desc=f"Scraping
{nombre_fuente}"):
        result = future.result()
        if result:
            noticias.extend(result)

print(f"{nombre_fuente} → Total publicaciones filtradas: {len(noticias)}")
registrar_log(nombre_fuente, len(noticias), errores, inicio)
return noticias


# ----------------------------------------
# FUENTES
# ----------------------------------------
def scrape_dinero():
    return scrape_generico("Dinero.com",
"https://www.dinero.com/buscador?query={keyword}&page={page}",
"https://www.dinero.com", "a[href*='/articulo/']", use_selenium=True)

def scrape_portafolio():
    return scrape_generico("Portafolio.co",
"https://www.portafolio.co/buscar/{keyword}/page={page}", "https://www.portafolio.co",
"a[href*='/noticias/']")

def scrape_larepublica():
    return scrape_generico("La República",
"https://www.larepublica.co/economia?query={keyword}&page={page}",
"https://www.larepublica.co", "a[href*='/economia/']", use_selenium=True)

def scrape_valoraanalitik():
    return scrape_generico("Valora Analitik",
"https://www.valoraanalitik.com/?s={keyword}&ggl={page}",
"https://www.valoraanalitik.com", "a[href*='/20']", use_selenium=True)
```

```
def scrape_eltiempo():
    return scrape_generico("El Tiempo",
"https://www.eltiempo.com/buscar/{keyword}/{page}", "https://www.eltiempo.com",
"a[href*='/noticia/']")

def scrape_elespectador():
    return scrape_generico("El Espectador",
"https://www.elespectador.com/buscar/?q={keyword}&page={page}",
"https://www.elespectador.com", "a[href*='/noticias/']")

def scrape_semana():
    return scrape_generico("Semana",
"https://www.semana.com/buscador/?query={keyword}&page={page}",
"https://www.semana.com", "a[href*='/articulo/']")

def scrape_investing():
    return scrape_generico("Investing",
"https://www.investing.com/search/?q={keyword}&tab=news&page={page}",
"https://www.investing.com", "a[href*='/news/']", use_selenium=False)

def scrape_superfinanciera():
    return scrape_generico("Superintendencia Financiera",
"https://www.superfinanciera.gov.co/inicio/comunicados-de-
prensa?s={keyword}&page={page}", "https://www.superfinanciera.gov.co",
"a[href*='/comunicados']", use_selenium=True)

def scrape_bvc():
    return scrape_generico("Bolsa de Valores de Colombia",
"https://www.bvc.com.co/mercado/buscar?q={keyword}&page={page}",
"https://www.bvc.com.co", "a[href*='/noticias']", use_selenium=True)

def scrape_bloomberg():
    return scrape_generico("Bloomberg",
"https://www.bloomberg.com/search?query={keyword}&page={page}",
"https://www.bloomberg.com", "a[href*='/news/']", use_selenium=True)

def scrape_tradingview():
    return scrape_generico("TradingView",
"https://www.tradingview.com/search/?query={keyword}&page={page}",
"https://www.tradingview.com", "a[href*='/ideas/']", use_selenium=True)

def scrape_yahoo_finance():
    return scrape_generico("Yahoo Finance",
"https://finance.yahoo.com/search?p={keyword}&fr2=p%3Afp%2Cm%3Asa&page={page
}", "https://finance.yahoo.com", "a[href*='/news/']", use_selenium=True)

def scrape_casadebolsa():
```

```
    return scrape_generico("Casa de Bolsa",
"https://www.casadebolsa.com.co/?s={keyword}&page={page}",
"https://www.casadebolsa.com.co", "a[href*='/noticias']", use_selenium=True)

# ----------------------------------------
# NEWSAPI
# ----------------------------------------
def buscar_noticias_newsapi(fechas, top_n=5):
    url = "https://newsapi.org/v2/everything"
    resultados = []
    query = "colcap OR 'bolsa de valores de colombia' OR ecopetrol OR bancolombia OR
mercado bursátil OR finanzas colombia"

    for fecha in tqdm(fechas, desc="Complementando con NewsAPI"):
        params = {
            "q": query,
            "from": str(fecha),
            "to": str(fecha),
            "sortBy": "relevancy",
            "pageSize": top_n,
            "apiKey": NEWSAPI_KEY
        }
        try:
            r = requests.get(url, params=params, timeout=10)
            print(f"NewsAPI - Fecha: {fecha}, Estado: {r.status_code}")
            r.raise_for_status()
            data = r.json()
            for articulo in data.get("articles", []):
                resultados.append({
                    "titulo": articulo["title"],
                    "url": articulo["url"],
                    "fuente": articulo["source"]["name"],
                    "contenido_articulo": articulo.get("description", ""),
                    "fecha_publicacion": articulo["publishedAt"][:10]
                })
        except requests.RequestException as e:
            logging.error(f"Error en NewsAPI para {fecha}: {e}")
    return resultados

# ----------------------------------------
# VERIFICACIÓN DE COBERTURA TEMPORAL
# ----------------------------------------
def verificar_cobertura(df):
    fechas_objetivo = pd.date_range("2022-01-01", "2024-12-31").date
    if "fecha_publicacion" not in df.columns:
        logging.error("Columna 'fecha_publicacion' no encontrada en el DataFrame.")
        print("❌ Error: El DataFrame no contiene la columna 'fecha_publicacion'.")
        return fechas_objetivo
```

```
    df["fecha_publicacion"] = df["fecha_publicacion"].fillna("No disponible")
    fechas_validas = df["fecha_publicacion"].str.match(r"\d{4}-\d{2}-\d{2}")
    fechas_scrapeadas = pd.to_datetime(df.loc[fechas_validas, "fecha_publicacion"],
errors="coerce").dt.date
    faltantes = sorted(set(fechas_objetivo) - set(fechas_scrapeadas.dropna()))

    print(f"\n❌ Días sin noticias: {len(faltantes)}")
    if faltantes:
        faltantes_df = pd.DataFrame(faltantes, columns=["fecha_sin_noticia"])
        faltantes_df.to_excel("dias_sin_noticias.xlsx", index=False)
        print("🗁 Archivo 'dias_sin_noticias.xlsx' generado con días faltantes.")
    return faltantes

# ----------------------------------------
# FUNCIÓN PRINCIPAL
# ----------------------------------------
def main():
    all_noticias = []
    funciones_scraping = [
        scrape_dinero
    ]

    os.makedirs(os.path.dirname(OUTPUT_PATH), exist_ok=True)
    with open(LOG_PATH, "w", encoding="utf-8") as f:
        f.write("LOG DE SCRAPING COLCAP ENERO 2022\n\n")

    for func in funciones_scraping:
        noticias = func()
        for noticia in noticias:
            if "fecha_publicacion" not in noticia:
                logging.warning(f"Artículo sin 'fecha_publicacion': {noticia.get('url', 'URL
desconocido')}")
        all_noticias.extend(noticias)

    if not all_noticias:
        print("⚠ No se encontraron noticias. Verifica las fuentes, selectores, clave de
NewsAPI o la configuración de ChromeDriver.")
        return

    df = pd.DataFrame(all_noticias)
    print(f"Total de noticias antes de eliminar duplicados: {len(df)}")
    df.drop_duplicates(subset=["url"], inplace=True)
    print(f"Total de noticias después de eliminar duplicados: {len(df)}")
    df.to_excel(OUTPUT_PATH, index=False)
    print(f"\n✅ Noticias guardadas en: {OUTPUT_PATH}")

    faltantes = verificar_cobertura(df)
    if faltantes:
```

```
        noticias_newsapi = buscar_noticias_newsapi(faltantes, top_n=5)
        if noticias_newsapi:
            df_extra = pd.DataFrame(noticias_newsapi)
            df = pd.concat([df, df_extra], ignore_index=True)
            df.drop_duplicates(subset=["url"], inplace=True)
            df.to_excel(OUTPUT_PATH, index=False)
            print("🔄 Archivo actualizado con noticias complementadas desde NewsAPI.")

if __name__ == "__main__":
    main()
```

2. Código para clasificación de sentimientos

```
import pandas as pd
import re
import nltk
nltk.download('punkt_tab')
from nltk.corpus import stopwords
from nltk.tokenize import word_tokenize
from nltk.stem import WordNetLemmatizer
from nltk.sentiment import SentimentIntensityAnalyzer
from IPython.display import display
#pip install textblob
from textblob import TextBlob


#Lectura de archivos
file_path = r"C:\Users\luisa\OneDrive - universidadean.edu.co\Trabajo final
especialización\Noticias 2.xlsx"
colcap_file_path = r"C:\Users\luisa\OneDrive - universidadean.edu.co\Trabajo final
especialización\MSCI COLCAP_202002-201502.csv"
df_noticias = pd.read_excel(file_path, sheet_name="Hoja1")
print(df_noticias.head())


# Diccionario para convertir nombres de meses en español a números
# Diccionarios para convertir nombres de meses en español a números
meses_es = {
    "enero": "01", "febrero": "02", "marzo": "03", "abril": "04",
    "mayo": "05", "junio": "06", "julio": "07", "agosto": "08",
    "septiembre": "09", "octubre": "10", "noviembre": "11", "diciembre": "12"
}

meses_abreviados = {
    "ENE": "01", "FEB": "02", "MAR": "03", "ABR": "04",
    "MAY": "05", "JUN": "06", "JUL": "07", "AGO": "08",
    "SEP": "09", "OCT": "10", "NOV": "11", "DIC": "12"
}
```

```
import requests
from bs4 import BeautifulSoup
from datetime import datetime


def obtener_fecha_noticia(url):
    try:
        response = requests.get(url, timeout=10)
        if response.status_code == 200:
            soup = BeautifulSoup(response.text, 'html.parser')

            # Buscar la fecha en formato tradicional en español
            fecha_match_es = re.search(r'(\d{1,2} de [a-zA-Z]+ de \d{4})', soup.text)
            if fecha_match_es:
                fecha_str = fecha_match_es.group(0)
                for mes, num in meses_es.items():
                    fecha_str = fecha_str.replace(f" de {mes} de ", f"/{num}/")
                fecha_dt = datetime.strptime(fecha_str, '%d/%m/%Y')
                return fecha_dt.strftime('%d/%m/%Y')

            # Buscar la fecha en formato ISO (YYYY-MM-DD)
            fecha_match_iso = re.search(r'(\d{4}-\d{2}-\d{2})', soup.text)
            if fecha_match_iso:
                fecha_dt = datetime.strptime(fecha_match_iso.group(0), '%Y-%m-%d')
                return fecha_dt.strftime('%d/%m/%Y')

            # Buscar la fecha en formato abreviado (Ej: 15 MAR 2025)
            fecha_match_abrev = re.search(r'(\d{1,2})\s([A-Z]{3})\s(\d{4})', soup.text)
            if fecha_match_abrev:
                dia, mes_abrev, anio = fecha_match_abrev.groups()
                mes_num = meses_abreviados.get(mes_abrev, "00")
                fecha_str = f"{dia}/{mes_num}/{anio}"
                fecha_dt = datetime.strptime(fecha_str, '%d/%m/%Y')
                return fecha_dt.strftime('%d/%m/%Y')

        # Extraer fecha de la URL si está en formato YYYY-MM-DD
        fecha_match_url = re.search(r'/(\d{4})-(\d{2})-(\d{2})/', url)
        if fecha_match_url:
            fecha_dt = datetime.strptime("-".join(fecha_match_url.groups()), '%Y-%m-%d')
            return fecha_dt.strftime('%d/%m/%Y')

    except Exception as e:
        print(f"Error al procesar {url}: {e}")
    return None

# Aplicar la función a todas las URLs
df_noticias['Fecha Publicación'] = df_noticias['url'].apply(obtener_fecha_noticia)
```

```
# Guardar el archivo con las fechas actualizadas
df_noticias.to_excel(r"C:\Users\luisa\OneDrive - universidadean.edu.co\Trabajo final especialización\Noticias_con_fechas.xlsx", index=False)


##%%%%%%%%%%%%%%%%%%%%%%%%%PROCESAMIENTO DE TEXTO PARA EL ANÁLISIS DE SENTIMIENTOS

# Descargar recursos necesarios de NLKT
nltk.download('punkt')
nltk.download('stopwords')
nltk.download('wordnet')
nltk.download('vader_lexicon')

# Iniciar lematizador y analizador de sentimiento
lemmatizer = WordNetLemmatizer()
sia = SentimentIntensityAnalyzer()

# Lista de stopwords en español
stop_words = set(stopwords.words('spanish'))

# Función para limpiar el texto
def limpiar_texto(texto):
    texto = texto.lower() # pasar todo  a minúsculas
    texto = re.sub(r'\d+', '', texto) # eliminar números
    texto = re.sub(r'\W', '', texto) # eliminar caracteres especiales
    tokens = word_tokenize(texto) #Tokenización
    tokens = [word for word in tokens if word not in stop_words] #eliminar stopwords
    tokens = [lemmatizer.lemmatize(word) for word in tokens] # Lematización
    return ''.join(tokens)

#####%%%%%%%%%%%%% CONSTRUCCIÓN DE MODELOS PARA CALIFICACIÓN DE SENTIMIENTOS

#MODELO 1 CON EL MÉTODO VADER
#Aplicar limpieza e texto a la columna del contenido del artículo#
#df_noticias['contenido_limpio'] =
df_noticias['contenido_articulo'].astype(str).apply(limpiar_texto)

# Aplicar análisis de sentimientos con VADER
#df_noticias['sentimiento'] = df_noticias['contenido_limpio'].apply(lambda x:
sia.polarity_scores(x)['compound'])

# Clasificar sentimiento en positivo / negativo
#df_noticias['sentimiento_clasificacion'] = df_noticias['sentimiento'].apply(lambda x:
'positivo' if x > 1 else ('negativo' if x < -0.05 else 'neutro'))

#mostrar resultados
```

```
#print("texto limpio")
#display(df_noticias)

#guardar archivo en excel
#df_noticias.to_excel(r"C:\Users\luisa\OneDrive - universidadean.edu.co\Trabajo final especialización\texto limpio.xlsx")

#Conclusión: el métdo VADER no es concluyente ya que está entrenado en inglés

#MODELO 2 MÉTODO TexBlob
# Función para analizar el sentimiento con TextBlob en español
def obtener_sentimiento(texto):
    blob = TextBlob(texto)
    polaridad = blob.sentiment.polarity  # Valor entre -1 (negativo) y 1 (positivo)
    if polaridad > 0.05:
        return 'positivo'
    elif polaridad < -0.05:
        return 'negativo'
    else:
        return 'neutro'

# Aplicar el análisis de sentimiento
df_noticias['sentimiento_clasificacion'] = df_noticias['contenido_articulo'].astype(str).apply(obtener_sentimiento)

# Mostrar resultados actualizados
print("texto limpio")
display(df_noticias)

#guardar archivo en excel
df_noticias.to_excel(r"C:\Users\luisa\OneDrive - universidadean.edu.co\Trabajo final especialización\texto limpio 2.xlsx")


#CRUCE DE NOTICIAS CON SENTIMIENTOS

# Cargar nuevamente el archivo con delimitador correcto
df_colcap = pd.read_csv(colcap_file_path, delimiter=';', encoding='latin1')

# Renombrar columnas para mayor claridad
df_colcap.columns = ["Fecha", "Valor Hoy", "Valor Ayer", "Variación Absoluta", "Variación Porcentual", "Variación 12 Meses", "Variación Año"]

# Convertir la columna de fecha a formato datetime
#df_colcap["Fecha"] = pd.to_datetime(df_colcap["Fecha"], format="%d/%m/%Y")
df_colcap.head()
# Convertir valores numéricos, eliminando caracteres como comas y %
for col in ["Valor Hoy", "Valor Ayer", "Variación Absoluta", "Variación Porcentual", "Variación 12 Meses", "Variación Año"]:
```

```
    df_colcap[col] = df_colcap[col].astype(str).str.replace(',', '').str.replace('%',
'').astype(float)

# Convertir la columna de fecha en df_noticias al mismo formato
#df_noticias["Fecha Publicación"] = pd.to_datetime(df_noticias["Fecha
Publicación"]).dt.strftime('%Y-%m-%d')
#df_noticias.head()
# Filtrar noticias con fecha válida
df_noticias = df_noticias.dropna(subset=["Fecha Publicación"])

# Asignar valores numéricos a los sentimientos para calcular promedio por fecha
sentiment_mapping = {"positivo": 1, "negativo": -1, "neutro": 0}
df_noticias["sentimiento_clasificacion"] =
df_noticias["sentimiento_clasificacion"].map(sentiment_mapping)

# Calcular el sentimiento promedio por fecha
df_sentimiento_diario = df_noticias.groupby("Fecha
Publicación")["sentimiento_clasificacion"].mean().reset_index()
df_sentimiento_diario.rename(columns={"sentimiento_clasificacion":
"sentimiento_clasificacion Promedio"}, inplace=True)
df_colcap.rename(columns={"Fecha": "Fecha Publicación"}, inplace=True)

df_colcap["Fecha Publicación"] = pd.to_datetime(df_colcap["Fecha Publicación"])
df_sentimiento_diario["Fecha Publicación"] =
pd.to_datetime(df_sentimiento_diario["Fecha Publicación"])

# Unir los datos de COLCAP con el sentimiento promedio de las noticias
df_merged = pd.merge(df_colcap, df_sentimiento_diario, on="Fecha Publicación",
how="left")

df_merged.to_excel(r"C:\Users\luisa\OneDrive - universidadean.edu.co\Trabajo final
especialización\merged.xlsx")
```