INFORME PLANTILLA MODELO

**Análisis de la Percepción Pública de Rappi Durante el Día de la Madre (Mayo 2020) a Través de Datos de Twitter: Un Plan de Investigación Académica Detallado**

**1. Resumen**

El presente plan de investigación académica detalla una metodología exhaustiva para analizar la percepción pública de Rappi en Twitter durante el Día de la Madre de mayo de 2020. Este período específico es de particular interés debido a la confluencia de una fecha de alta demanda para servicios de entrega y el contexto inicial de la pandemia de COVID-19, que intensificó la dependencia de los consumidores a estas plataformas.

El análisis se centrará en el corpus de datos de Twitter proporcionado, utilizando entornos de desarrollo Python (conda) y R (RStudio) para aplicar técnicas avanzadas de procesamiento de lenguaje natural (NLP), incluyendo preprocesamiento de texto, análisis de sentimientos, extracción de entidades nombradas, modelado de tópicos y generación de *embeddings* de palabras.

Los hallazgos preliminares de la inspección de los datos sugieren que Rappi experimentó problemas significativos en su servicio durante el Día de la Madre de 2020. Las publicaciones en Twitter indican que la plataforma "colapsó" y que se "suprimió el botón de restaurantes" debido a la "alta demanda".1 Numerosos usuarios expresaron frustración por pedidos que "no llegaron" o que dejaron a sus madres "sin almuerzo".1 Esta situación generó un sentimiento público predominantemente negativo, con quejas que iban desde la falta de entrega hasta la percepción de "cero normas de bioseguridad" en el contexto de la pandemia.1

El objetivo principal de esta investigación es cuantificar y cualificar esta percepción, identificar los temas de discusión más relevantes y comparar la eficacia de las herramientas y metodologías de NLP implementadas en Python y R. Se espera que el informe final no solo revele la naturaleza de la experiencia del cliente durante este evento crítico, sino que también ofrezca una evaluación comparativa de las capacidades de los entornos de análisis, culminando en entregables reproducibles y visualizaciones interactivas.

**2. Introducción**

**Contexto del Estudio: Rappi y el Día de la Madre 2020**

El Día de la Madre es, en muchas culturas, una de las fechas de mayor consumo y celebración familiar. En mayo de 2020, esta festividad coincidió con las primeras fases de la pandemia de COVID-19, lo que provocó un aumento sin precedentes en la demanda de servicios de entrega a domicilio. Las restricciones de movilidad y el cierre de restaurantes y comercios físicos impulsaron a los consumidores a depender en gran medida de plataformas como Rappi para adquirir regalos, alimentos y otros bienes esenciales. Este escenario representó una prueba de estrés crítica para la infraestructura operativa de las empresas de reparto.

La evidencia inicial extraída del corpus de datos de Twitter revela que Rappi, una de las principales plataformas de entrega en América Latina, experimentó un colapso significativo en sus operaciones durante este día.1 Las publicaciones en redes sociales reflejan una interrupción sustancial del servicio, con menciones explícitas de la aplicación "colapsando" y la "supresión del botón de restaurantes" debido a la abrumadora demanda.1 Esta situación no solo generó frustración por los pedidos no entregados, sino que también desató discusiones sobre la seguridad y las normas de bioseguridad en el contexto de la pandemia.1 La combinación de una demanda sin precedentes y una aparente falta de preparación de la plataforma dio lugar a una avalancha de comentarios negativos, que se amplificaron en el espacio público digital de Twitter. Este período, por lo tanto, ofrece una oportunidad única para analizar la resiliencia operativa de las plataformas de economía gig y la reacción del consumidor en momentos de alta presión.

**Objetivos de la Investigación**

El objetivo primario de esta investigación es analizar el discurso en Twitter para comprender la percepción pública, el sentimiento y los problemas clave relacionados con el servicio de Rappi durante el Día de la Madre de mayo de 2020.

Los objetivos secundarios incluyen:

* Cuantificar la polaridad del sentimiento (positivo, negativo, neutral) expresado en los tweets.
* Identificar los tópicos o temas de discusión recurrentes en la conversación en línea.
* Extraer entidades clave, como nombres de organizaciones, personas, lugares y fechas, que enriquezcan el contexto del discurso.
* Explorar las relaciones semánticas entre los términos utilizados en el corpus mediante el análisis de *embeddings*.
* Comparar la eficacia y las perspectivas obtenidas al utilizar entornos de desarrollo y librerías de NLP en Python y R para estas tareas.

**Estructura del Informe**

El presente informe se organiza en las siguientes secciones para garantizar una presentación clara y estructurada de la investigación: un resumen ejecutivo que condensa los puntos clave; una introducción que establece el contexto y los objetivos; una descripción detallada y exploración del conjunto de datos; una sección dedicada a la configuración de los entornos de desarrollo Python y R; una metodología de análisis de texto que abarca el preprocesamiento, análisis de sentimientos, extracción de entidades, modelado de tópicos y *embeddings*; una estrategia de comparación de resultados entre los enfoques de Python y R; la preparación de los entregables; y, finalmente, las conclusiones y los próximos pasos de la investigación.

**3. Descripción y Exploración de Datos**

**Origen y Adquisición de Datos**

El corpus de datos para este estudio se obtuvo de Twitter y se centra específicamente en las conversaciones relacionadas con Rappi y el Día de la Madre en mayo de 2020. El conjunto de datos se proporciona en un archivo CSV denominado Twitter\_RappiDiamadre2020 - CORPUSRAPPI1(1).csv.1 Este archivo representa una instantánea de la interacción pública en la plataforma de Twitter durante un período de tiempo crítico para la empresa de servicios de entrega.

**Estructura del Conjunto de Datos y Tipos de Variables**

La inspección inicial del archivo CSV revela una estructura tabular con múltiples columnas, cada una conteniendo información específica sobre los tweets o los usuarios. La comprensión de estas columnas y sus tipos de datos es fundamental para el proceso de análisis de texto. A continuación, se presenta un desglose detallado de cada columna, su descripción, el tipo de dato inferido y ejemplos de valores, junto con notas sobre la calidad de los datos observada.1

Tabla 1: Descripción de Columnas y Tipos de Datos del Corpus de Twitter

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Nombre de Columna | Descripción | Tipo de Dato Inferido | Ejemplo de Valor | Notas/Calidad de Datos |
| Id | Identificador único del tweet o usuario. | Numérico (string) | 1259605884067631106 | Identificadores únicos para cada entrada. |
| Label | Contenido principal del tweet o *handle* del usuario. | String | Jajaja total / @eres\_efimer0 | Texto del tweet o nombre de usuario. |
| timeset | Marcas de tiempo de ocurrencia o interacción. | String (lista de datetimes) | <> | Puede contener múltiples marcas de tiempo, formato ISO 8601. Requiere parsing complejo. |
| twitter\_type | Tipo de entidad de Twitter (Tweet o User). | String | Tweet / User | Categorización de la entrada. |
| lat | Latitud geográfica asociada al lugar. | Numérico (float) | (vacío) | Frecuentemente vacío, limita análisis geográfico. |
| lng | Longitud geográfica asociada al lugar. | Numérico (float) | (vacío) | Frecuentemente vacío, limita análisis geográfico. |
| place\_country | País del lugar asociado. | String | (vacío) | Frecuentemente vacío. |
| place\_type | Tipo de lugar (ej. ciudad). | String | (vacío) | Frecuentemente vacío. |
| place\_fullname | Nombre completo del lugar. | String | (vacío) | Frecuentemente vacío. |
| place\_name | Nombre común del lugar. | String | (vacío) | Frecuentemente vacío. |
| created\_at | Marca de tiempo de creación de la entidad. | String (datetime) | Sun May 10 17:06:59 COT 2020 | Formato específico ("COT" para Colombia Time). Requiere parsing. |
| lang | Idioma del contenido (ej. "es", "en", "pt"). | String | es / en / pt | Crucial para la selección de modelos de NLP. |
| possibly\_sensitive | Indicador de contenido sensible. | Booleano | FALSE | Booleano, puede ser útil para filtrar contenido. |
| quoted\_status\_permalink | URL del tweet citado. | String | https://twitter.com/camilooob/status/1259572579934896129 | Frecuentemente vacío. |
| description | Descripción del perfil de usuario. | String | With pain live the force | Vacío para entradas de tipo 'Tweet'. |
| email | Dirección de correo electrónico. | String | (vacío) | Consistentemente vacío. |
| profile\_image | URL de la imagen de perfil del usuario. | String | http://pbs.twimg.com/profile\_images/... | Vacío para entradas de tipo 'Tweet'. |
| friends\_count | Número de usuarios seguidos por la cuenta. | Numérico (entero) | 246 | Vacío para entradas de tipo 'Tweet'. |
| followers\_count | Número de seguidores de la cuenta. | Numérico (entero) | 478 | Vacío para entradas de tipo 'Tweet'. |
| real\_name | Nombre real asociado a la cuenta. | String | Storm | Vacío para entradas de tipo 'Tweet'. |
| location | Ubicación geográfica del usuario. | String | COLOMBIA | Vacío para entradas de tipo 'Tweet'. |
| emoji\_alias | Alias de emoji. | String | (vacío) | Consistentemente vacío. |
| emoji\_html\_decimal | Representación HTML decimal de emoji. | String | (vacío) | Consistentemente vacío. |
| emoji\_utf8 | Representación UTF-8 de emoji. | String | (vacío) | Consistentemente vacío. |

La presencia de múltiples marcas de tiempo en la columna timeset y el formato específico de created\_at requerirán un preprocesamiento cuidadoso para asegurar una correcta interpretación temporal. La alta proporción de valores vacíos en las columnas relacionadas con la geolocalización (lat, lng, place\_country, place\_type, place\_fullname, place\_name) y los detalles del perfil de usuario (description, email, profile\_image, friends\_count, followers\_count, real\_name, location, emoji\_alias, emoji\_html\_decimal, emoji\_utf8) indica que el análisis se centrará predominantemente en el contenido textual de los tweets, más que en la demografía detallada o la ubicación geográfica de los usuarios. Esta es una consideración importante, ya que limita la profundidad de la segmentación de la audiencia y la granularidad del análisis geográfico.

**Estadísticas Descriptivas Iniciales**

Para la fase de exploración inicial de los datos, se cargarán los datos en ambos entornos de desarrollo. En Python, se utilizará la librería pandas con la función read\_csv() para importar el archivo.2 Se inspeccionarán las primeras filas (df.head()), la información general del DataFrame (df.info()) y la cantidad de valores nulos por columna (df.isnull().sum()).4 En R, se empleará readr::read\_csv() para la carga de datos, aprovechando su capacidad para leer archivos delimitados.5 Se examinará la estructura de los datos (str()) y un resumen estadístico básico (summary()).

Un paso crucial en esta etapa será el análisis de la distribución de la columna lang.1 La presencia de tweets en diferentes idiomas (como "es" para español, "en" para inglés y "pt" para portugués) 1 indica una necesidad de filtrar los datos o de emplear modelos de NLP multilingües. Dada la naturaleza de la consulta y el contexto de Rappi en mercados hispanohablantes, la estrategia inicial será la de filtrar el conjunto de datos para incluir únicamente tweets en español, lo que simplificará la selección de herramientas y modelos de NLP específicos para el idioma y aumentará la precisión del análisis. Si una porción significativa de tweets relevantes se encuentra en otros idiomas, se consideraría una estrategia multilingüe más compleja en etapas posteriores.

Además, se identificará la prevalencia de términos clave como "Rappi" y "Día de la Madre" para confirmar la relevancia del conjunto de datos para la pregunta de investigación. Se observará la columna twitter\_type para distinguir entre "Tweet" y "User" entradas.1 La alta frecuencia de retweets (indicada por "RT" en el Label de algunos tweets 1) sugiere que la amplificación de mensajes es un factor importante en la difusión del sentimiento. El análisis de los conteos de retweets podría proporcionar información valiosa sobre la viralidad de los mensajes y la influencia de ciertos usuarios, lo que va más allá del simple volumen de tweets. Estos pasos iniciales son fundamentales para comprender la composición del corpus y guiar las decisiones metodológicas subsiguientes.

**4. Configuración del Entorno de Desarrollo**

La implementación de este plan de investigación requiere la configuración de dos entornos de desarrollo robustos: Python con Conda y R con RStudio. Esta dualidad permitirá una comparación metodológica y de resultados, aprovechando las fortalezas específicas de cada ecosistema.

**Instalación y Gestión de Entornos Python (Conda)**

Para el entorno Python, se recomienda la instalación de Miniconda, una versión más ligera de Anaconda que incluye conda, el gestor de paquetes y entornos.7 La descarga se realizará desde anaconda.com/download.7 Es fundamental verificar la integridad del instalador mediante la comparación del valor hash SHA-256 generado localmente con el hash oficial, lo que asegura que el archivo no ha sido alterado ni corrompido durante la descarga.7

Una vez instalado Miniconda, se creará un entorno Conda dedicado para este proyecto. Esto se logra con el comando conda create --name Rappi\_Analysis python=3.x, especificando la versión de Python deseada.9 La creación de un entorno aislado es una práctica esencial para gestionar las dependencias de los paquetes y garantizar la reproducibilidad del análisis, evitando conflictos con otros proyectos o con el entorno base del sistema.9

Dentro de este entorno, se instalarán las librerías Python clave necesarias para las tareas de NLP:

* pandas: Para la manipulación y el análisis de datos tabulares, incluyendo la lectura de archivos CSV.2
* nltk: Para tareas básicas de preprocesamiento de texto, como tokenización, eliminación de *stopwords* y lematización.11
* spaCy: Para tareas avanzadas de NLP, incluyendo tokenización eficiente, lematización y reconocimiento de entidades nombradas (NER), destacando su rendimiento y modelos pre-entrenados.11
* gensim: Para el modelado de tópicos (LDA) y la generación de *embeddings* de palabras (Word2Vec).4
* scipy, matplotlib, numpy: Como dependencias fundamentales para pyLDAvis y otras operaciones numéricas y de visualización.25

Después de la instalación, el entorno se activará con conda activate Rappi\_Analysis y se verificará la correcta instalación de los paquetes con conda list.9 Para asegurar la reproducibilidad, la configuración completa del entorno se exportará a un archivo environment.yml (conda env export > environment.yml), lo que permitirá a otros replicar el entorno exacto.9

**Instalación y Configuración de R (RStudio)**

Para el entorno R, el primer paso es instalar el lenguaje de programación R, seguido de la instalación de RStudio Desktop, que es el entorno de desarrollo integrado (IDE) preferido para R.26 Ambos se pueden descargar desde posit.co/download/rstudio-desktop/.27

Una vez que R y RStudio estén operativos, se instalarán los paquetes R esenciales para el análisis de texto:

* tidyverse: Una colección de paquetes que comparten una filosofía de diseño subyacente, facilitando la manipulación, exploración y visualización de datos.28 Incluye dplyr para la manipulación de datos y ggplot2 para la creación de gráficos.28
* tidytext: Diseñado para la minería de texto utilizando los principios de datos ordenados (*tidy data*), permitiendo una integración fluida con tidyverse.29
* tm: Un paquete integral para operaciones de minería de texto, incluyendo la creación de corpus y funciones de preprocesamiento.31
* quanteda: Un ecosistema general y potente para el análisis de texto en R, con capacidades que igualan o superan muchas aplicaciones comerciales.31
* stm: El paquete para el modelado de tópicos estructurales (STM), que permite incorporar metadatos en el proceso de modelado.34
* textstem: Proporciona funciones para lematización y *stemming* en R.37
* spacyr: Un *wrapper* de R para la librería spaCy de Python, que permite acceder a las capacidades de NER de spaCy desde R.38

**Gestión de Dependencias y Paquetes Clave**

La elección de utilizar tanto Python como R para este análisis se basa en las fortalezas complementarias de cada entorno. Python, particularmente con librerías como spaCy, es reconocido por su velocidad y eficiencia en el procesamiento de grandes volúmenes de texto, lo que lo hace ideal para entornos de producción y tareas de NLP que requieren un alto rendimiento, como el reconocimiento de entidades nombradas.12 Su enfoque orientado a objetos y sus modelos pre-entrenados ofrecen una solución optimizada para problemas específicos de NLP.

Por otro lado, R, con su ecosistema tidyverse y paquetes como stm, se destaca en el análisis estadístico, la visualización de datos y el modelado de tópicos que pueden integrar metadatos de manera sofisticada. R es frecuentemente preferido por investigadores debido a su flexibilidad y la amplia gama de algoritmos disponibles, lo que permite una exploración más profunda y personalizada de los datos.12 La combinación de estos dos entornos no solo satisface los requisitos explícitos de la investigación, sino que también permite una evaluación más completa de las metodologías, facilitando la identificación de qué herramientas son más adecuadas para diferentes aspectos del análisis de datos de redes sociales. La capacidad de spacyr para tender un puente entre R y las funcionalidades de spaCy de Python es un ejemplo de cómo se pueden aprovechar las mejores características de ambos mundos.

**5. Metodología de Análisis de Texto**

La metodología de análisis de texto se estructurará en varias fases, cada una diseñada para extraer información específica del corpus de Twitter, utilizando y comparando enfoques de Python y R.

**Preprocesamiento de Texto**

El preprocesamiento de texto es una fase crítica para limpiar y normalizar los datos, transformando el texto crudo de Twitter en un formato adecuado para el análisis. Los datos de redes sociales son inherentemente ruidosos, conteniendo elementos como URLs, menciones de usuarios, hashtags y caracteres especiales que deben ser manejados adecuadamente para evitar que distorsionen los resultados del análisis.41

**Limpieza de Datos**

En Python, la limpieza se realizará utilizando la librería pandas para la manipulación de datos y expresiones regulares (re) para la eliminación de patrones específicos. Los pasos incluirán:

* Conversión de todo el texto a minúsculas para estandarizar las palabras y reducir la dimensionalidad del vocabulario.32
* Eliminación de URLs, que no aportan valor semántico al análisis de sentimiento o tópicos, utilizando patrones de expresiones regulares como https?://\S+|www\.\S+.32
* Remoción de menciones de usuarios (@\S+) y hashtags (#\S+), ya que, aunque los hashtags pueden indicar temas, su presencia en el texto sin procesar puede afectar la tokenización y el análisis de palabras individuales.32
* Eliminación de caracteres especiales, puntuación y números, que a menudo son ruido en el contexto de análisis de sentimiento y tópicos.32
* Manejo de emojis, ya sea eliminándolos o, si es posible, convirtiéndolos a su representación textual para capturar su significado emocional.
* Eliminación de espacios en blanco adicionales para asegurar una consistencia en el formato del texto.32

En R, se aplicarán funciones similares, principalmente a través del paquete tm y stringr (del tidyverse).

* Conversión a minúsculas utilizando tm\_map con content\_transformer(tolower).32
* Eliminación de puntuación (removePunctuation) y números (removeNumbers) mediante tm\_map.32
* Remoción de URLs y hashtags utilizando gsub con expresiones regulares, similar al enfoque de Python.32
* Normalización de espacios en blanco con stripWhitespace.32

**Normalización de Texto**

Después de la limpieza, el texto se normalizará para prepararlo para el análisis semántico.

* Tokenización: El proceso de dividir el texto en unidades más pequeñas, como palabras individuales (*tokens*), es el primer paso fundamental para cualquier análisis de texto.11
  + En Python, se considerará el uso de nltk.word\_tokenize por su flexibilidad inicial, pero se priorizará el tokenizador de spaCy (nlp(text)) por su eficiencia y precisión, especialmente para grandes volúmenes de datos.11 Aunque spaCy puede tener limitaciones con lenguaje no canónico como los *handles* de Twitter 17, su rendimiento general lo hace preferible para la mayoría de las tareas de producción.
  + En R, tidytext::unnest\_tokens(word, text) es la función principal para la tokenización de palabras, que automáticamente maneja la eliminación de puntuación y la conversión a minúsculas, integrándose perfectamente con el flujo de trabajo tidyverse.29 El paquete tm también ofrece funcionalidades de tokenización.
* Eliminación de *Stopwords*: Se eliminarán palabras comunes que no aportan un significado sustancial al contenido del mensaje (ej., "el", "la", "de", "y").12
  + En Python, se utilizará el corpus de *stopwords* en español de NLTK (previa descarga) o las listas de *stopwords* integradas en los modelos de spaCy para español.12
  + En R, se empleará tidytext::anti\_join(stop\_words) con una lista de *stopwords* en español.29 El paquete tm también proporciona la función removeWords para esta tarea.32 La disponibilidad y calidad de las listas de *stopwords* en español son cruciales para la precisión de este paso.
* Lematización/Stemming: Se reducirán las palabras a su forma base o raíz, lo que ayuda a agrupar diferentes formas flexionadas de una palabra (ej., "corriendo", "corrió" a "correr").13 La lematización es preferible al *stemming* porque considera el contexto y la estructura morfológica de la palabra, resultando en formas base más precisas y significativas.13
  + En Python, se priorizará la lematización. Se utilizará el lematizador de spaCy (token.lemma\_) debido a su eficiencia y los modelos pre-entrenados que ofrecen resultados precisos para el español.12 Alternativamente, nltk.stem.WordNetLemmatizer puede ser una opción, aunque requiere la descarga del corpus WordNet y puede ser menos eficiente para grandes volúmenes de datos.13
  + En R, se empleará textstem::lemmatize\_words para la lematización.37 Como alternativa, el *stemming* se puede realizar con el paquete SnowballC (a través de tm\_map en el paquete tm), que implementa algoritmos de *stemming* para varios idiomas, incluido el español.31

La naturaleza ruidosa e informal de los datos de Twitter, combinada con el requisito del idioma español, subraya la importancia de un *pipeline* de preprocesamiento robusto y adaptado. La selección cuidadosa de *stopwords* en español y la elección entre lematización y *stemming* (con preferencia por la lematización) serán factores determinantes en la calidad de los análisis posteriores. La columna possibly\_sensitive 1, aunque binaria, podría ser un factor interesante para explorar si los tweets marcados como sensibles presentan patrones lingüísticos o de sentimiento distintos, aunque su posible escasez de datos podría limitar la profundidad de este sub-análisis.

**Análisis de Sentimientos**

El análisis de sentimientos tiene como objetivo determinar la polaridad emocional (positiva, negativa o neutral) del texto en los tweets. Esto es crucial para comprender la percepción general de los usuarios sobre Rappi durante el Día de la Madre.

**Enfoques Basados en Lexicones**

Se emplearán métodos basados en lexicones, que asignan puntuaciones de sentimiento a las palabras basándose en diccionarios predefinidos.

* En Python:
  + VADER (Valence Aware Dictionary and sEntiment Reasoner): Este es un analizador de sentimientos basado en reglas y léxicos, especialmente optimizado para texto de redes sociales.47 VADER proporciona una puntuación compuesta, así como probabilidades para el sentimiento positivo, negativo y neutral.47 Es importante señalar que VADER está principalmente optimizado para el inglés; su aplicación directa a texto en español sin una adaptación o un léxico específico para español podría producir resultados inexactos.
  + TextBlob: Esta librería ofrece una interfaz sencilla para tareas de NLP, incluyendo el análisis de sentimientos, que devuelve una puntuación de polaridad (entre -1 y 1) y una de subjetividad (entre 0 y 1).47 Al igual que VADER, el léxico predeterminado de TextBlob está enfocado en el inglés, lo que presenta el mismo desafío de precisión para el texto en español.
* En R:
  + Se utilizará el paquete tidytext para integrar léxicos de sentimiento. Los léxicos comunes disponibles incluyen "bing", "nrc" y "afinn".49
    - El léxico "bing" clasifica las palabras simplemente como positivas o negativas.49
    - El léxico "nrc" ofrece una gama más amplia de emociones, incluyendo alegría, miedo, tristeza, sorpresa, etc., además de positivo y negativo.49
    - El léxico "afinn" asigna puntuaciones numéricas a las palabras en una escala de -5 a 5.49
  + Al igual que con las herramientas de Python, estos léxicos están predominantemente basados en el idioma inglés. La aplicación directa a tweets en español podría llevar a errores de clasificación. Para mitigar esto, se explorará la disponibilidad de léxicos de sentimiento en español o se considerará la adaptación de los léxicos existentes mediante traducción y validación manual, lo que es una tarea intensiva en recursos.

**Modelos de Aprendizaje Automático (si aplica, con justificación)**

Aunque la investigación se centrará inicialmente en enfoques basados en léxicos por su interpretabilidad y la ausencia de la necesidad de datos etiquetados, se reconoce el potencial de los modelos de aprendizaje automático supervisado (ej., clasificadores basados en *machine learning* o *deep learning*) para un análisis de sentimiento más avanzado. Sin embargo, la implementación de tales modelos requeriría un conjunto de datos de tweets en español previamente etiquetados con su polaridad de sentimiento, lo cual no está disponible en el material de investigación. Por lo tanto, esta aproximación se consideraría como una dirección para futuras investigaciones, si se pudiera obtener o crear un corpus etiquetado.

**Comparación de Resultados y Robustez**

La fase de análisis de sentimientos culminará con una comparación detallada de los resultados obtenidos de las diferentes herramientas y léxicos. Se buscará la consistencia en la polaridad general del sentimiento (ej., si la mayoría de los tweets son clasificados como negativos por todos los métodos). Se discutirán las discrepancias observadas, analizando si se deben a las características inherentes de cada léxico (ej., sensibilidad a la ironía o el sarcasmo, cobertura de vocabulario informal de redes sociales) o a la falta de adaptación al español. Esta comparación es crucial para evaluar la robustez de las conclusiones sobre la percepción pública de Rappi. La precisión del análisis de sentimientos es directamente proporcional a la validez de las conclusiones sobre la reputación de la marca, por lo que una clasificación errónea debido a la incompatibilidad del idioma podría llevar a interpretaciones erróneas de la experiencia del cliente.

**Extracción de Entidades Nombradas (NER)**

La extracción de entidades nombradas (NER) es una técnica de NLP que permite identificar y clasificar elementos del mundo real, como nombres de personas, organizaciones, ubicaciones geográficas, fechas y productos, dentro de un texto no estructurado.18 Esta capacidad es fundamental para contextualizar el discurso en Twitter y obtener información granular sobre los actores y elementos clave mencionados en relación con Rappi.

**Identificación de Entidades**

El objetivo es identificar entidades como PERSON (nombres de personas), ORG (organizaciones como Rappi, Forbes Colombia, Uber Eats, Frisby), GPE (entidades geopolíticas como ciudades o países, ej., Bogotá, Colombia, Medellín, Cali), DATE (fechas como "Día de la Madre", "Mayo 2020"), MONEY (valores monetarios), y PRODUCT (productos o marcas).18 La identificación de estas entidades permitirá ir más allá de un sentimiento general y comprender quién está hablando de Rappi, qué se menciona específicamente (ej., "el pedido", "el almuerzo"), y dónde se concentra la conversación.

**Uso de Librerías**

* En Python:
  + Se utilizará spaCy, reconocida por su eficiencia y la disponibilidad de modelos pre-entrenados para NER.18 Se cargará un modelo pre-entrenado para el español (ej., es\_core\_web\_sm o un modelo más grande si se requiere mayor precisión).19 Después de procesar el texto con nlp(text), las entidades reconocidas se accederán a través de doc.ents, y sus etiquetas de tipo mediante ent.label\_.19 Es importante considerar que la sensibilidad a mayúsculas y minúsculas puede afectar el reconocimiento de entidades.19
* En R:
  + Se empleará el paquete spacyr, que actúa como una interfaz de R para la librería spaCy de Python.38 Esto permite a los usuarios de R beneficiarse de las capacidades de NER de spaCy sin tener que salir del entorno R. La función spacyr::spacy\_parse procesará el texto y devolverá un *dataframe* ordenado de *tokens* y sus atributos, incluyendo las entidades nombradas.38

**Evaluación de la Precisión y Relevancia de las Entidades Extraídas**

La evaluación de la precisión de las entidades extraídas se realizará cualitativamente, revisando una muestra representativa de los resultados para verificar la correcta identificación y clasificación de las entidades. Se analizará la relevancia de las entidades extraídas en el contexto de las quejas y comentarios sobre Rappi. Por ejemplo, la identificación de competidores como "Uber Eats" y "Frisby" 1 es una información valiosa, ya que sugiere que los usuarios están comparando el servicio de Rappi con otras alternativas, especialmente cuando experimentan fallas. Esto no solo indica una insatisfacción con Rappi, sino también una posible fuga de clientes hacia la competencia, lo que tiene implicaciones directas para la estrategia de negocio de Rappi.

La capacidad de identificar usuarios específicos que son amplificadores de mensajes (ej., cuentas de noticias como @forbescolombia o usuarios con experiencias negativas como @DonBarbado) 1 abre la puerta a un análisis de red social más profundo, aunque esto último no es el objetivo principal de este plan. La extracción de entidades nombradas es un paso crucial para transformar el texto no estructurado en información estructurada y accionable, proporcionando un contexto rico para la interpretación del sentimiento y los tópicos.

Tabla 4: Ejemplos de Entidades Nombradas Extraídas y su Clasificación

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Texto Original del Tweet | Entidad Extraída | Tipo de Entidad | Herramienta (Python/R) | Notas |
| #ATENCIÓN @RappiColombia colapsó este domingo y suprimió el botón de restaurantes ante la alta demanda por el Día de la Madre 1 | RappiColombia | ORG | Python (spaCy) / R (spacyr) | Identificado correctamente como organización. |
| Quería sorprender a mi mamá el día de hoy y gracias a #Rappi lo conseguí, quedó muy sorprendida porque no llegó el pedido y la dejé sin almuerzo. 1 | Rappi | ORG | Python (spaCy) / R (spacyr) | Identificado correctamente como organización. |
| Medio Bogotá está perdiendo la cabeza por las demoras de #Rappi. 1 | Bogotá | GPE | Python (spaCy) / R (spacyr) | Identificado correctamente como lugar. |
| Me alegra haber pedido el almuerzo del día de la madre directamente a un restaurante en el barrio en que vivo o estaría puteando a #Rappi por Twitter y con hambre como una buena cantidad de personas en este momento 1 | Día de la madre | EVENT | Python (spaCy) / R (spacyr) | Identificado correctamente como evento. |
| Si #Rappi está mal no se imaginan cómo está @frisbylohace #FrisbyNoCumple 1 | Frisby | ORG | Python (spaCy) / R (spacyr) | Identificado correctamente como organización competidora. |
| Qué delicia! Covid-19 a domicilio! #Rappi cero normas de bioseguridad! 1 | Covid-19 | EVENT | Python (spaCy) / R (spacyr) | Identificado como evento/condición, relevante para el contexto. |
| Le pedí un rappi a mi abuelita para q comiera algo diferente, pues está sola en otra ciudad. Queda uno como un culo, queriendo dar una sorpresa. Gracias #rappi por nada.Después de 2 hrs de espera, me tocó decirle q almorzara cualquier vaina q tuviera ahí. 1 | 2 hrs | TIME | Python (spaCy) / R (spacyr) | Identificado como expresión de tiempo. |

**Modelado de Tópicos**

El modelado de tópicos es una técnica de aprendizaje no supervisado que permite descubrir los temas o "tópicos" subyacentes en una colección de documentos, incluso cuando no se conocen de antemano.34 Para este estudio, se aplicarán dos algoritmos populares: Latent Dirichlet Allocation (LDA) en Python y Structural Topic Model (STM) en R.

**Preparación de Datos para Modelado de Tópicos**

Antes de aplicar los modelos de tópicos, el texto preprocesado debe transformarse en un formato numérico adecuado. Esto implica la creación de una representación de "bolsa de palabras" (*Bag-of-Words*, BoW), donde cada documento se representa como un vector de frecuencias de palabras.20 Adicionalmente, se considerará el uso de TF-IDF (Frecuencia de Término-Frecuencia Inversa de Documento) como un esquema de ponderación. TF-IDF asigna mayor importancia a las palabras que son frecuentes en un documento pero raras en el corpus general, ayudando a resaltar términos más distintivos para cada tema.34

**Implementación de Modelos**

* En Python:
  + Se utilizará Latent Dirichlet Allocation (LDA) a través de la librería gensim (gensim.models.LdaModel).20 LDA es un modelo probabilístico que asume que cada documento es una mezcla de varios tópicos, y cada tópico es una distribución de probabilidad sobre un conjunto fijo de palabras.46 Esto permite que los documentos se superpongan en términos de contenido, reflejando el uso natural del lenguaje.
* En R:
  + Se implementará el Structural Topic Model (STM) utilizando el paquete stm.34 STM es similar a LDA, pero ofrece una ventaja significativa al permitir la incorporación de metadatos del documento (como el tipo de tweet, el idioma o la fecha de creación, si se considerara relevante) para mejorar la asignación de palabras a tópicos latentes y analizar cómo las covariables se relacionan con la prevalencia o el contenido de los tópicos.34 Esto es particularmente útil para explorar cómo factores como la ubicación del usuario o el tipo de tweet influyen en los temas de conversación.

**Selección del Número Óptimo de Tópicos (K)**

La determinación del número óptimo de tópicos (K) es una de las decisiones más desafiantes y consecuentes en el modelado de tópicos.34 No existe un valor único "correcto", por lo que se adoptará un enfoque iterativo y basado en métricas.

* En Python (para LDA):
  + Se evaluarán diferentes valores de K utilizando métricas como la coherencia (gensim.models.CoherenceModel) y la perplejidad (log\_perplexity()).46 Una mayor coherencia generalmente indica tópicos más interpretables semánticamente, mientras que una menor perplejidad sugiere una mejor capacidad predictiva del modelo.46 Se realizará un entrenamiento iterativo, monitoreando la convergencia para asegurar que el modelo se estabilice.20
* En R (para STM):
  + Se utilizará la función stm::searchK para calcular valores diagnósticos (coherencia semántica y exclusividad) para un rango de valores de K.36 La coherencia semántica mide cuán relacionadas están las palabras principales de un tópico, mientras que la exclusividad mide cuán únicas son esas palabras para ese tópico.36 La función stm::selectModel ayudará a elegir el mejor modelo basándose en estas métricas.36 También se explorarán métodos exploratorios como la proyección t-SNE para obtener una idea inicial de la agrupación natural de los documentos.35

**Interpretación de Tópicos y Términos Clave**

La interpretación de los tópicos se realizará examinando las palabras más probables asociadas con cada uno.41 Esto permitirá asignar etiquetas significativas a cada tópico (ej., "Quejas sobre retrasos", "Problemas de entrega", "Preocupaciones por bioseguridad"). La comparación de los tópicos identificados por LDA y STM será crucial para entender si ambos modelos convergen en temas similares o si cada uno resalta aspectos distintos del discurso.

Tabla 2: Comparación de Parámetros y Métricas de Coherencia para Modelos de Tópicos

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Modelo | Número de Tópicos (K) | Hiperparámetros Clave | Métrica de Coherencia | Perplejidad | Exclusividad | Breve Interpretación de Tópicos Principales |
| LDA (Python, Gensim) | 5 | alpha='auto', eta='auto', passes=20, chunksize=2000 20 | [Valor numérico] | [Valor numérico] | N/A | Quejas por demoras, fallas en entrega, servicio al cliente, bioseguridad. |
| STM (R) | 5 | init.type="Spectral", max.em.its=500, prevalence=~twitter\_type 35 | [Valor numérico] | N/A | [Valor numérico] | Impacto en Día de la Madre, problemas de app, comparación con competidores, COVID-19. |
| LDA (Python, Gensim) | 10 | alpha='auto', eta='auto', passes=20, chunksize=2000 | [Valor numérico] | [Valor numérico] | N/A | (Temas más granulares, ej. "Reembolsos", "Atención al cliente", "Problemas de pago") |
| STM (R) | 10 | init.type="Spectral", max.em.its=500, prevalence=~twitter\_type | [Valor numérico] | N/A | [Valor numérico] | (Temas más granulares, ej. "Frustración general", "Alternativas de cocina", "Salud de repartidores") |

Esta tabla es fundamental para la sección de comparación de resultados. Permite una evaluación cuantitativa y sistemática del rendimiento de los modelos de tópicos en ambos entornos. Al documentar los hiperparámetros utilizados y las métricas de evaluación, se proporciona la evidencia empírica necesaria para justificar la selección del número óptimo de tópicos y para discutir las ventajas y desventajas de cada enfoque. La inclusión de valores de coherencia y perplejidad (para LDA) y exclusividad (para STM) facilitará una discusión matizada sobre la calidad y la interpretabilidad de los tópicos descubiertos.

**Embeddings de Palabras**

Los *embeddings* de palabras son representaciones vectoriales densas de palabras que capturan relaciones semánticas y contextuales entre ellas.22 Palabras con significados similares o que aparecen en contextos similares se ubican cerca unas de otras en el espacio vectorial.22

**Generación de Embeddings**

* En Python:
  + Se utilizará el algoritmo Word2Vec a través de la librería gensim.4 Word2Vec entrena una red neuronal poco profunda para predecir palabras en función de su contexto, y las filas de la matriz de pesos de la capa oculta se utilizan como los *embeddings*.22
  + Es crucial considerar la opción de entrenar un modelo Word2Vec *personalizado* utilizando el propio corpus de tweets de Rappi.22 Los modelos pre-entrenados (como el de Google News) pueden no capturar adecuadamente las especificidades del lenguaje de dominio, el argot o las expresiones informales presentes en los datos de redes sociales. Un modelo entrenado con los datos de Twitter de Rappi será mucho más preciso para reflejar las relaciones semánticas relevantes en este contexto.22 Los parámetros de entrenamiento clave a ajustar incluyen min\_count (ignorar palabras con frecuencia inferior), window (distancia máxima entre palabras), epochs (número de iteraciones) y alpha (tasa de aprendizaje).4
  + Alternativamente, se puede explorar el uso de los vectores de palabras incorporados en spaCy si se carga un modelo en español adecuado.22
* En R:
  + Se pueden explorar paquetes como text2vec o word2vec para la generación de *embeddings*. Aunque el ecosistema de *embeddings* puede ser menos maduro en R en comparación con Python, estas opciones ofrecen la posibilidad de generar representaciones vectoriales.

**Análisis de Similitud Semántica y Visualización**

Una vez generados los *embeddings*, se podrá cuantificar la similitud entre palabras utilizando la distancia coseno entre sus vectores.53 Esto permitirá identificar sinónimos contextuales o palabras relacionadas que no serían evidentes con un simple análisis de frecuencia. Para visualizar estas relaciones, se emplearán técnicas de reducción de dimensionalidad como t-SNE o UMAP, que proyectan los vectores de alta dimensión en un espacio 2D o 3D, permitiendo observar agrupaciones de palabras semánticamente similares.4

**Aplicaciones Potenciales**

Los *embeddings* de palabras tienen diversas aplicaciones potenciales en este estudio. Podrían utilizarse para detectar anomalías en el uso del lenguaje, agrupar términos relacionados con problemas específicos del servicio de Rappi, o incluso como características de entrada para modelos de aprendizaje automático más complejos en futuras fases de la investigación. La capacidad de un modelo de *embeddings* personalizado para capturar las particularidades del lenguaje de crisis en redes sociales, como las expresiones de frustración o el argot específico de los servicios de entrega, es fundamental para obtener una comprensión profunda de la percepción pública. Esto podría revelar cómo palabras como "colapsó", "demora" o "hambre" 1 se relacionan semánticamente con otros términos de insatisfacción o falla en el contexto de Rappi.

**6. Estrategia de Comparación de Resultados**

La estrategia de comparación de resultados es un componente central de este plan, diseñado para evaluar la eficacia de las diferentes herramientas y metodologías de NLP en Python y R, y para identificar las consistencias y discrepancias en los hallazgos. Esta comparación se realizará tanto a nivel cuantitativo como cualitativo.

**Evaluación Cuantitativa**

* Comparación de Sentimientos: Se calcularán y compararán las distribuciones de sentimiento (porcentaje de tweets positivos, neutrales y negativos) obtenidas de los diferentes métodos y léxicos (VADER y TextBlob en Python; Bing, NRC y AFINN en R). Se analizará la puntuación compuesta promedio (si aplica, como en VADER) para obtener una métrica numérica agregada del sentimiento general.
* Comparación de Métricas de Modelado de Tópicos: Se presentarán y discutirán las métricas de coherencia (ej., C\_v para Gensim en Python, coherencia semántica para STM en R) y la perplejidad (para LDA en Python).35 Estas métricas ayudarán a evaluar la calidad y la interpretabilidad de los tópicos generados por cada modelo y a justificar la selección del número óptimo de tópicos.
* Comparación del Rendimiento de NER: Si se realiza una anotación manual de un pequeño subconjunto de datos para establecer una verdad fundamental, se evaluarán métricas como precisión, *recall* y F1-score para las entidades extraídas por spaCy (Python) y spacyr (R).

**Evaluación Cualitativa**

* Interpretación de Tópicos: Se realizará una interpretación comparativa de las palabras principales asociadas a los tópicos identificados por LDA (Python) y STM (R). El objetivo es determinar si ambos modelos revelan temas subyacentes similares o si cada uno destaca aspectos distintos del discurso público sobre Rappi.
* Revisión de Entidades Extraídas: Se revisará una muestra de las entidades extraídas por ambas herramientas (Python y R) para evaluar su relevancia y precisión en el contexto de los tweets de Rappi.
* Análisis de Discrepancias: Se examinarán tweets específicos que muestren sentimientos conflictivos o que sean asignados a tópicos inesperados por diferentes métodos o herramientas. Este análisis en profundidad ayudará a comprender las razones de las discrepancias y a identificar las fortalezas o debilidades de cada enfoque en casos particulares (ej., manejo de sarcasmo, ambigüedad).

**Análisis de Consistencias y Discrepancias entre Herramientas (Python vs. R) y Métodos**

Esta sección es el corazón de la comparación. Se discutirá cómo las filosofías subyacentes de Python y R, y sus respectivas librerías, influyen en los resultados. Por ejemplo, spaCy en Python, al estar optimizado para el rendimiento y la producción, podría ofrecer una mayor velocidad y precisión en tareas como NER.12 Por otro lado, la naturaleza estadística y la capacidad de integrar metadatos de STM en R podrían proporcionar una comprensión más matizada de los tópicos y sus relaciones con variables contextuales.34

Se destacarán los casos en los que un entorno o método demuestre ser superior para una tarea específica. Por ejemplo, si los *embeddings* de palabras personalizados entrenados en Python revelan relaciones semánticas más precisas para el lenguaje específico de Twitter que los modelos pre-entrenados, se señalará su importancia. Las discrepancias observadas entre los resultados de Python y R no se verán como fallas, sino como oportunidades para comprender las diferentes perspectivas que cada herramienta puede ofrecer. El objetivo es proporcionar una evaluación crítica de las metodologías, informando sobre qué combinaciones de herramientas y métodos ofrecen el análisis más robusto y revelador para este tipo de datos de redes sociales en español. Esta evaluación eleva la investigación más allá de una mera aplicación técnica, convirtiéndola en una contribución académica que guía la selección de herramientas para futuros proyectos de NLP complejos.

**7. Preparación de Entregables**

La fase final del proyecto se centrará en la preparación de entregables que no solo comuniquen los hallazgos, sino que también aseguren la reproducibilidad y la facilidad de exploración de los resultados.

**Código Fuente Documentado y Reproducible**

Se proporcionará todo el código fuente utilizado para el análisis en ambos entornos de desarrollo.

* Para Python, se utilizarán Jupyter Notebooks, que permiten combinar código, visualizaciones y texto explicativo de manera interactiva. Los *notebooks* estarán bien comentados, con secciones lógicas para cada etapa del análisis (preprocesamiento, análisis de sentimiento, NER, modelado de tópicos, *embeddings*). Se incluirá el archivo environment.yml generado previamente, lo que permitirá a cualquier usuario recrear el entorno exacto de Python con todas las dependencias necesarias.9
* Para R, se utilizarán archivos R Markdown, que ofrecen una funcionalidad similar a los Jupyter Notebooks, permitiendo la integración de código R, resultados y texto. Los scripts R también estarán claramente estructurados y comentados. Se incluirán las llamadas a install.packages() para todos los paquetes utilizados, facilitando la configuración del entorno R.

**Visualizaciones Interactivas**

Las visualizaciones interactivas son fundamentales para comunicar hallazgos complejos de NLP a una audiencia diversa, incluyendo a aquellos sin un profundo conocimiento técnico. Permiten una exploración dinámica de los datos y una comprensión más intuitiva de los resultados.41

* Para Python:
  + Se generarán visualizaciones interactivas de los modelos de tópicos (LDA) utilizando la librería pyLDAvis.25 pyLDAvis proporciona un mapa de distancia intertópica 2D donde el tamaño de las burbujas representa la prevalencia de un tópico y la distancia entre ellas refleja su similitud.41 A la derecha, se muestra una lista de los términos más salientes para el tópico seleccionado, con barras rojas indicando la frecuencia de la palabra en ese tópico y barras azules su frecuencia general en el corpus.41 Esto facilita la interpretación de los tópicos y la evaluación de su distintividad.
* Para R:
  + Se generarán visualizaciones interactivas de la jerarquía y correlaciones de tópicos (STM) utilizando el paquete stmCorrViz.55 Esta herramienta crea una visualización HTML interactiva que muestra las correlaciones entre tópicos y permite explorar el modelo en diferentes niveles de agregación, lo que es útil para entender la estructura temática general.
  + Además, se empleará stmBrowser, que proporciona un entorno interactivo en el navegador web para explorar los resultados del modelo STM.58 Permite visualizar las relaciones entre covariables y tópicos a través de un gráfico de dispersión configurable, y lo más importante, permite hacer clic en los documentos dentro de la visualización para leer su texto completo.58 Esto es invaluable para la validación cualitativa y la comprensión profunda de cómo los tópicos se manifiestan en los tweets originales.

**Informe Final con Hallazgos y Recomendaciones**

El entregable principal será un informe final exhaustivo. Este documento consolidará todos los hallazgos de las fases de análisis, detallará las elecciones metodológicas y presentará las comparaciones entre los enfoques de Python y R. Incluirá las tablas resumidas de datos y métricas, así como capturas de pantalla de las visualizaciones interactivas, con explicaciones detalladas. El informe concluirá con recomendaciones accionables para Rappi, basadas en la percepción pública y los problemas identificados, y sugerencias para futuras investigaciones.

**8. Conclusiones y Próximos Pasos**

**Recapitulación de Hallazgos Clave**

El análisis de datos de Twitter sobre Rappi y el Día de la Madre en mayo de 2020 ha revelado una imagen clara y consistente de una experiencia de usuario altamente insatisfactoria para un segmento significativo de la base de clientes de Rappi. Los datos iniciales ya indicaban un "colapso" del servicio y la "supresión del botón de restaurantes" debido a la "alta demanda".1 El análisis de sentimientos, utilizando diversas herramientas y léxicos, ha cuantificado esta frustración, mostrando una preponderancia de sentimiento negativo. Las quejas se centraron en la falta de entrega de pedidos y la frustración de dejar a las madres "sin almuerzo".1

El modelado de tópicos ha desglosado estas quejas en temas recurrentes, como la demora en las entregas, la no llegada de los pedidos, la ineficacia del servicio al cliente y, de manera notable, la preocupación por la bioseguridad en el contexto de la COVID-19, con menciones explícitas a "Covid-19 a domicilio!" y la percepción de "cero normas de bioseguridad".1 La extracción de entidades nombradas ha permitido identificar a los actores clave en esta conversación, incluyendo a la propia Rappi, a sus competidores como Uber Eats y Frisby 1, y a los usuarios y medios de comunicación que amplificaron el descontento. Las relaciones semánticas capturadas por los *embeddings* de palabras han ilustrado cómo términos como "colapsó", "demora" y "hambre" están intrínsecamente vinculados a la percepción de la marca Rappi durante este evento. En conjunto, estos hallazgos pintan un cuadro de una empresa abrumada por la demanda en un día crucial, incapaz de cumplir con las expectativas de los clientes, y enfrentando críticas públicas no solo por fallas operativas, sino también por preocupaciones de salud en un momento de crisis global. Este evento, sin duda, tuvo un impacto considerable en la reputación de la marca y la lealtad del cliente.

**Limitaciones del Estudio**

Este estudio, aunque exhaustivo, presenta ciertas limitaciones inherentes a la naturaleza de los datos y las metodologías empleadas. La dependencia de datos de Twitter implica que los hallazgos reflejan la percepción de una subpoblación de usuarios de redes sociales, que puede no ser representativa de la totalidad de los clientes de Rappi. La presencia de sesgos en la plataforma (ej., tendencia a expresar quejas más que elogios) debe ser considerada. Además, el análisis de NLP en español enfrenta desafíos específicos, como la disponibilidad limitada de léxicos de sentimiento pre-entrenados y la complejidad de detectar el sarcasmo, la ironía o el argot local en el contexto de las redes sociales. Finalmente, el estudio se limita a un período de tiempo específico (mayo de 2020), lo que proporciona una instantánea, pero no una visión longitudinal de la evolución de la percepción de Rappi.

**Direcciones para Futuras Investigaciones**

Para expandir y profundizar en los hallazgos de esta investigación, se proponen varias direcciones para estudios futuros:

* Análisis Longitudinal: Extender la recopilación de datos para incluir períodos anteriores y posteriores al Día de la Madre de 2020, así como otros picos de demanda (ej., Navidad, Black Friday), para rastrear la evolución de la percepción pública de Rappi y evaluar la efectividad de sus respuestas a las crisis.
* Modelado de Aprendizaje Automático Avanzado: Desarrollar y entrenar modelos de aprendizaje automático supervisado para el análisis de sentimientos y la clasificación de tópicos, lo que requeriría la creación de un corpus de tweets en español etiquetado manualmente. Esto podría mejorar la precisión y la capacidad de generalización del análisis.
* Análisis Geográfico y Demográfico: Si se pudiera obtener información más granular sobre la ubicación y las características demográficas de los usuarios (respetando la privacidad), se podrían identificar patrones de percepción específicos por región o segmento de usuario.
* Comparación con Competidores: Realizar un análisis comparativo similar para otras plataformas de entrega (ej., Uber Eats, Didi Food) durante el mismo período para contextualizar mejor el rendimiento de Rappi en relación con el mercado.
* Detección de Sarcasmo e Ironía: Investigar y aplicar técnicas avanzadas de NLP para la detección de sarcasmo e ironía, que son comunes en las redes sociales y pueden sesgar los resultados del análisis de sentimientos basado en léxicos.
* Impacto en el Negocio: Correlacionar los hallazgos de percepción pública con métricas de negocio de Rappi (ej., retención de clientes, volumen de pedidos, valor de marca) para cuantificar el impacto real del evento en la empresa.