Universidad de Costa Rica

Escuela de Matemáticas

Análisis de datos I

CA-0411

Bitácoras

Análisis de sentimientos en mensajes de WhatsApp

PROFESOR

Joshua Cervantes Artavia

ESTUDIANTES

Gustavo Alberto Amador Fonseca - C20451

Luis Fernando Amey Apuy - C20470

Javier Hernández Navarro - C13674

I - 2025

Índice

1. Bitácora 1

Durante el I-Ciclo del 2024, en el curso de *Herramientas de Ciencia de Datos II*, se desarrolló un proyecto enfocado en el análisis de conversaciones grupales de WhatsApp y Telegram utilizando Python. El objetivo principal fue explorar las capacidades de las bibliotecas de procesamiento de lenguaje natural para extraer patrones de comunicación y evaluar los sentimientos expresados en los mensajes. Este proyecto dio a conocer las posibilidades del análisis computacional para entender las dinámicas de comunicación presentes en las conversaciones digitales, despertando un interés en el área.

El programa implementado en dicho proyecto permitió procesar datos crudos, realizar limpieza y preprocesamiento de texto, para así aplicar algoritmos de análisis de sentimientos mediante herramientas como *NLTK*, *TextBlob* y *VADER*. Se examinaron métricas como la frecuencia de mensajes, el uso de multimedia (stickers, imágenes) y la polaridad emocional (positiva, negativa o neutral) en dos contextos distintos: un chat grupal de la generación de actuariales y el chat grupal oficial del curso. Entre los hallazgos más relevantes se destacan la identificación de los miembros más activos, los días de mayor interacción y las tendencias emocionales predominantes en el grupo.

En este nuevo proyecto se propone un desafío técnico más ambicioso: desarrollar un algoritmo de clasificación propio que replique los análisis de sentimientos sin depender de las librerías especializadas ya existentes. La iniciativa busca construir desde cero un modelo de machine learning capaz de clasificar el tono emocional de los mensajes en positivos, negativos o neutrales, mediante el uso de un diccionario palabras etiquetadas, el diseño de un sistema de ponderación personalizado y el desarrollo del mecanismo de clasificación final.

El proyecto no solo permitirá comprender en profundidad los fundamentos del análisis de sentimientos, sino que también ofrecerá la oportunidad de comparar el rendimiento de nuestro modelo

con los resultados obtenidos mediante las herramientas convencionales. A lo largo de esta bitácora, se documentará minuciosamente cada fase de desarrollo, los obstáculos encontrados y los aprendizajes obtenidos en el proceso de la creación del algoritmo.

1.1. Descripción de la Tabla de Datos

1.1.1. Características de la tabla

La base de datos a utilizar puede extraerse de cualquier conversación en línea de las aplicaciones de WhatsApp y Telegram. Es importante recalcar que cada aplicación exporta los mensajes de una manera diferente, por lo que es necesario tratar cada una de ellas para unificar el formato y así facilitar el análisis.

Al exportar un chat de WhatsApp, se obtiene una base de datos en formato *.txt* que viene originalmente separada con saltos de línea y cada observación contiene la fecha, la hora, el autor y el contenido del mensaje. El código del proyecto anterior se encarga de separar dichos datos y los pone en las columnas identificadas de manera automatizada, es decir, tan solo basta darle dicho archivo crudo y el programa lo depura automáticamente.

En el caso de Telegram, al exportar un chat se obtiene una base de datos en formato *.json*, la cual tiene más variables que WhatsApp, como por ejemplo: *id*, *from_id* o *type*. De forma general, se redujeron las variables hasta llegar a una base de datos similar a la de WhatsApp, para mayor legibilidad y para tener uniformidad, ya que el enfoque del proyecto va más que todo hacia los mensajes de texto. Esto a su vez lo realiza el módulo de manera automatizada.

Ahora bien, como se mencionó anteriormente, se dispone de un módulo que es capaz de depurar las bases de datos crudas y dejarlas listas para ser utilizadas de forma autónoma. Dicho módulo recibe los archivos en *.txt* o *.json* y devuelve una base de datos con 5 variables: día, hora, autor, mensaje y editado, y cada observación corresponde a un mensaje del chat (esto se explicará a detalle más

adelante). Aunque las bases de datos de Telegram tienen más información por mensaje, en el proyecto anterior se decidió unificar el formato de ambas tablas para así tener mayor consistencia en los análisis, dejando únicamente las variables más relevantes para el enfoque dado.

Para este proyecto, se va a utilizar como base de datos únicamente el grupo de WhatsApp de la generación de estudiantes de Ciencias Actuariales de la Universidad de Costa Rica, esto debido a que se tiene un enfoque diferente, el cual consiste en utilizar los modelos que se verán en clase para replicar el análisis de sentimientos desde un enfoque más elemental.

1.1.2. Población de estudio

La base de datos contiene a 31 estudiantes de la generación de Ciencias Actuariales de la Universidad de Costa Rica. Para efectos de este proyecto, se procede a censurar los nombres de cada autor y se utilizará un id para cada autor.

1.1.3. Muestra observada

Se tomaron los mensajes de texto del grupo de la generación desde su creación el día 1/11/2023, hasta el día 2/4/2025. La misma está compuesta por 3698 observaciones y 5 variables.

1.1.4. Unidad estadística o individuos

Cada observación corresponde a un mensaje de texto enviado por un estudiante en el grupo de WhatsApp de la generación.

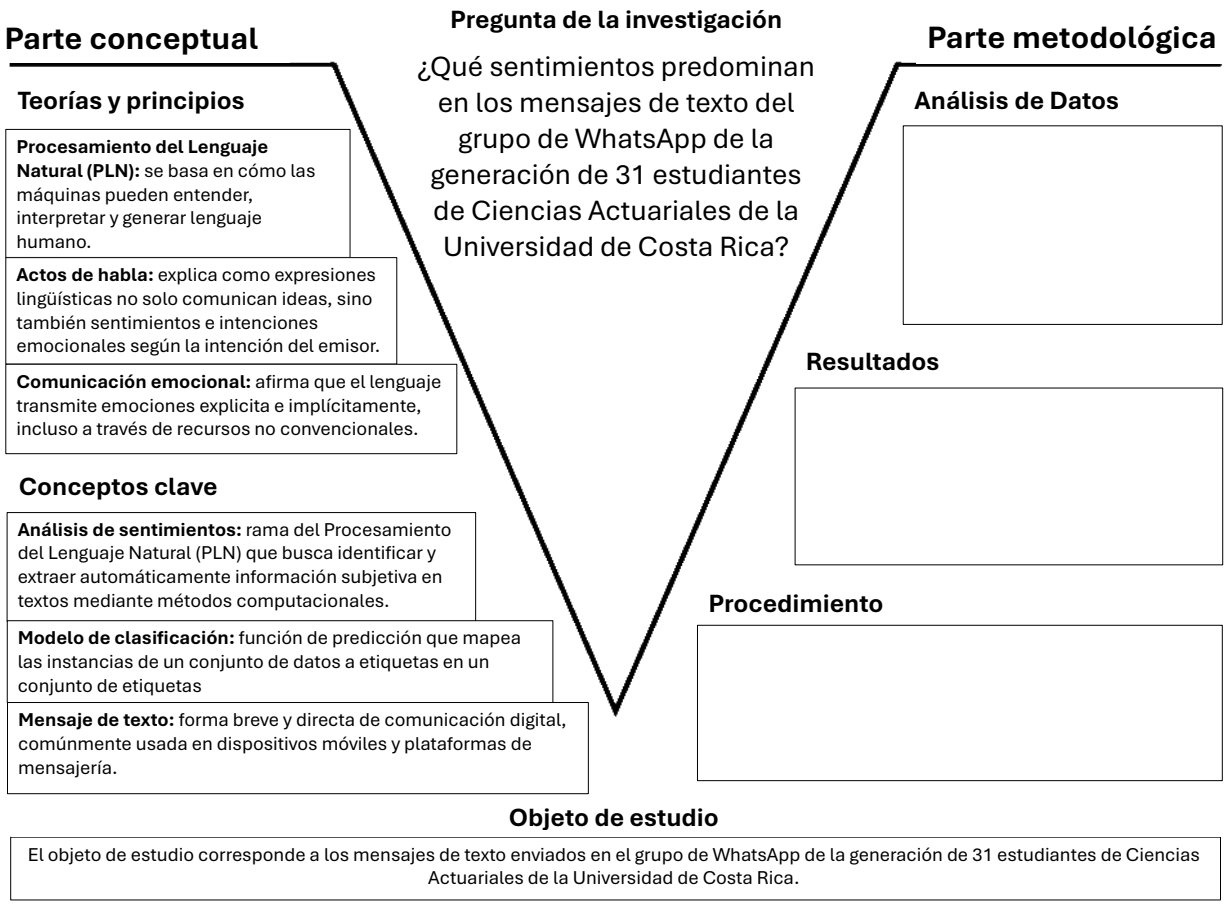
1.1.5. Variables de estudio

- **Día** (dia, double(date)): Día en que se envió el mensaje.
- **Hora** (hora, double): Hora en la que se envió el mensaje.

- **Autor** (autor, character): Autor del mensaje.
- **Mensaje** (mensaje, character): Contenido del mensaje enviado. Si el usuario envía una imagen, sticker, audio o documento; en el mensaje saldrá que el respectivo ha sido omitido, porque se está analizando texto. En caso de tratarse de los mensajes del sistema, estos aparecerán de manera regular, solamente que el autor será el nombre del grupo.
- **Editado** (editado, logical): Indica si el mensaje fue editado.

1.2. UVE de Gowin

Figura 1: Diagrama Uve de Gowin.



Fuente: Elaboración propia.

1.3. Pregunta de Investigación

¿Cómo se pueden emplear los algoritmos de clasificación en los mensajes de WhatsApp del grupo de la generación de 31 estudiantes de Ciencias Actuariales de la Universidad de Costa Rica para el análisis de sentimientos?

1.4. Objeto de Estudio

Analizar los mensajes de WhatsApp del grupo de la generación de 31 estudiantes de Ciencias Actuariales de la Universidad de Costa Rica, mediante algoritmos de clasificación, con el fin de analizar los sentimientos presentes en la comunicación escrita.

1.5. Conceptos Fundamentales

- **Algoritmos de clasificación:**

  Los algoritmos de clasificación son una modalidad de aprendizaje automático que segmenta los puntos de datos en grupos preestablecidos llamados clases. Los clasificadores son un modelo predictivo que adquiere atributos de clase a partir de los datos de entrada y aprende a asignarle posibles clases a los nuevos datos basándose en estos atributos adquiridos. En la ciencia de datos, los algoritmos de clasificación se emplean extensamente para anticipar patrones y resultados. En realidad, poseen una amplia gama de aplicaciones en la vida real, como la categorización de pacientes según posibles peligros sanitarios y el filtrado de correo no deseado. (Murel and Kavlakoglu, 2024)

- **Mensaje de WhatsApp:**

  Un mensaje de WhatsApp es una unidad de comunicación escrita enviada a través de la aplicación WhatsApp, una plataforma gratuita de mensajería instantánea que permite el inter-

cambio de textos, audios, imágenes, videos y otros archivos en tiempo real mediante conexión a internet. (Meta Platforms, Inc., 2025)

A diferencia de los mensajes de texto tradicionales (SMS), los mensajes de WhatsApp no están limitados por número de caracteres y permiten funciones interactivas como reacciones, reenvíos, respuestas encadenadas y cifrado de extremo a extremo. Esta forma de comunicación ha transformado las dinámicas sociales, académicas y laborales, facilitando la inmediatez y frecuencia en la interacción digital. (Church and de Oliveira, 2013)

- **Análisis de Sentimientos:**

  El análisis de sentimientos es una rama del Procesamiento del Lenguaje Natural (PLN) que busca identificar, extraer y clasificar automáticamente las emociones, opiniones o actitudes expresadas en un texto. Su objetivo principal es determinar si el contenido expresa una polaridad positiva, negativa o neutra hacia un tema específico. Este enfoque es ampliamente utilizado en diversas aplicaciones, desde análisis de opiniones en redes sociales hasta evaluación de comentarios en plataformas digitales. (Sande, 2018).

  Una de las definiciones más extendidas sobre el análisis de sentimientos y que abarca de manera más general fue descrita por Sande de la siguiente manera: “Conjunto de técnicas computacionales para la extracción, clasificación, comprensión y evaluación de opiniones expresadas en fuentes publicadas en Internet, comentarios en portales web y en otros contenidos generados por usuarios” (Sande, 2018).

1.6. Teorías Fundamentales

- **Teoría del Procesamiento del Lenguaje Natural (PLN):**

  Esta teoría se basa en cómo las máquinas pueden entender, interpretar y generar lenguaje humano. Utiliza técnicas como la tokenización, análisis semántico y sintáctico para extraer

significado del texto.

El libro *Speech and Language Processing* de Jurafsky y Martin proporciona una base extensa para entender la Teoría del Procesamiento del Lenguaje Natural. Los autores mencionan que el PLN es un campo interdisciplinario que combina lingüística, informática e inteligencia artificial para permitir que las máquinas procesen, entiendan y generen lenguaje humano: "el objetivo del PLN es desarrollar algoritmos y modelos computacionales que puedan analizar y generar lenguaje humano de manera efectiva" (Jurafsky and Martin, 1999).

Además, en el libro se describe que el PLN integra conocimientos lingüísticos y algoritmos computacionales para abordar desafíos como la ambigüedad o la variabilidad del lenguaje humano. Los autores Jurafsky y Martin declaran que: "El PLN avanza hacia sistemas que no solo procesen, sino que comprendan el lenguaje en toda su complejidad" (Jurafsky and Martin, 1999).

Por lo tanto, siguiendo esta idea se llega a que la base de aplicaciones como traductores automáticos, asistentes virtuales y cualquier elemento tecnológico que necesite comprender el lenguaje humano están basados en esta teoría.

- **Teoría de los Actos de Habla:**

  En *How to Do Things with Words* (1962), J. L. Austin sostiene que al hablar o escribir, muchas veces no solo describimos algo, sino que realizamos una acción mediante el lenguaje. A este tipo de expresiones las llama *performative utterances* (enunciados performativos), y se caracterizan porque al decirlas, el acto mismo se ejecuta. En el libro se menciona el ejemplo de "deberíamos decir que, al pronunciar estas palabras, estamos haciendo algo —es decir, casándonos— en lugar de informar que nos estamos casando" (Austin, 1962). Esta frase muestra como para Austin ciertos enunciados no describen una acción, sino que la constituyen en sí mismos.

Austin distingue entre tres niveles en todo acto lingüístico: el acto locutivo, el acto ilocutivo y el acto perlocutivo. En el contexto de los mensajes escritos, como los de WhatsApp o Telegram, lo fundamental es identificar el acto ilocutivo, ya que refleja lo que el emisor quiere hacer con sus palabras, de manera que exprese afecto, formule una queja, de una orden, entre otras (Austin, 1962).

Por último, Austin introduce la noción de felicidad o infelicidad de un acto de habla. Un mensaje no es simplemente válido o inválido por su contenido, sino por si se cumplen las condiciones adecuadas para que el acto tenga efecto. Acorde con la idea se menciona que "además de la emisión de las palabras del llamado enunciado performativo, muchas otras cosas deben, por regla general, estar bien y salir bien para que podamos decir que hemos realizado exitosamente nuestra acción" (Austin, 1962, p.14, traducción propia).

Por el contrario, para el caso de "infelicidad" se plantea que "cuando el enunciado resulta fallido, el procedimiento que pretendemos invocar es rechazado o mal ejecutado; y nuestra acción [...] queda anulada o sin efecto" (Austin, 1962, p.16, traducción propia).

De esta manera el autor introduce de manera clara el marco conceptual de los actos de habla, no se clasifican como verdaderos o falsos, sino como felices (happy) o infelices (unhappy), dependiendo de si se cumplen ciertas condiciones para que el acto se realice con éxito. Esta teoría permite clasificar los mensajes según el tipo de acción comunicativa que representan, clave para el análisis de sentimientos y emociones.

- **Teoría de la Comunicación Emocional:**

  Según Mohammad, el autor del artículo *Sentiment Analysis: Automatically Detecting Valence, Emotions, and Other Affectual States from Text*, el análisis de sentimientos va más allá de clasificar textos como positivos, negativos o neutros. Abarca también la detección de emociones específicas, la intensidad afectiva y las actitudes hacia temas o entidades. Esta mirada más

rica encaja con una concepción amplia de la comunicación emocional, donde el lenguaje no solo transmite información, sino también estados afectivos y sociales que reflejan la experiencia subjetiva del emisor (Mohammad, 2021).

El autor destaca que muchas veces las emociones no se expresan directamente, sino de forma implícita mediante metáforas, ironías, repeticiones o signos como emojis y puntuaciones exageradas, y señala que:

> En los textos en redes sociales están plagados de términos que no aparecen en los diccionarios comunes, incluyendo errores ortográficos (parlament), palabras con ortografía creativa (happeee), palabras con etiquetas (#loveumom), emoticones, abreviaturas (lmao), etc. Muchos de estos términos transmiten emociones. (Mohammad, 2021)

En el contexto actual esto es especialmente común en plataformas como WhatsApp o Telegram, donde el lenguaje informal y creativo introduce señales emocionales difíciles de detectar mediante enfoques tradicionales.

1.7. Primeras 5 Filas de la Tabla

Tabla 1: Tabla de mensajes.

día	**hora**	**autor**	**mensaje**	**editado**
2023-11-01	15:41:34	Autor 0	Los mensajes y las llamadas están cifrados ...	FALSE
2023-11-01	15:41:34	Autor 1	Autor 1 creó este grupo.	FALSE
2023-11-10	13:33:53	Autor 0	Autor 1 te añadió.	FALSE
2023-11-10	13:45:01	Autor 2	Se eliminó este mensaje.	FALSE
2023-11-10	19:48:35	Autor 1	Qué vieron en Interés? Estuvo muy feo?	FALSE

Fuente: Elaboración propia con la base de datos.

En la Tabla 1, se puede observar el formato final que se le dio a la base de datos en el proyecto anterior. Se realizó una pequeña pero sutil modificación en la variable *autor* para censurar los nombres reales de los autores, esto con el fin de mantener la integridad de dichos participantes.

Además, en la tabla se puede apreciar la variable de interés del proyecto, esta corresponde a *mensaje*. En cada observación, esta variable posee el contenido de dicho mensaje, el cual será utilizado para el análisis de sentimientos que se realizará en este trabajo.

Sin embargo, para este análisis de sentimientos se debe depurar aún más esta base de datos puesto que hay "mensajes" que no son relevantes, como la creación del grupo o cualquier documento, imagen o sticker adjunto omitido. Al mismo tiempo, en la mayoría de mensajes se cuenta con palabras que tampoco son de interés para dicho análisis, como lo es el caso de las preposiciones o también el uso de emojis. Este tratamiento será explicado más adelante.

Al probar el modelo con esta base de datos ya "limpia", se podrá desglosar de manera más

detallada los resultados concretos para cada autor.

1.8. Resumen de 5 Números

Tabla 2: Resumen de 5 Números de las Variables Cuantitativas.

	dia	**hora**
Min.	2023-11-01	01:12:12
1st Qu.	2024-01-23	13:12:27
Median	2024-05-22	15:24:52
Mean	2024-05-22	15:48:51
3rd Qu.	2024-08-21	19:46:35
Max.	2025-03-31	24:59:32

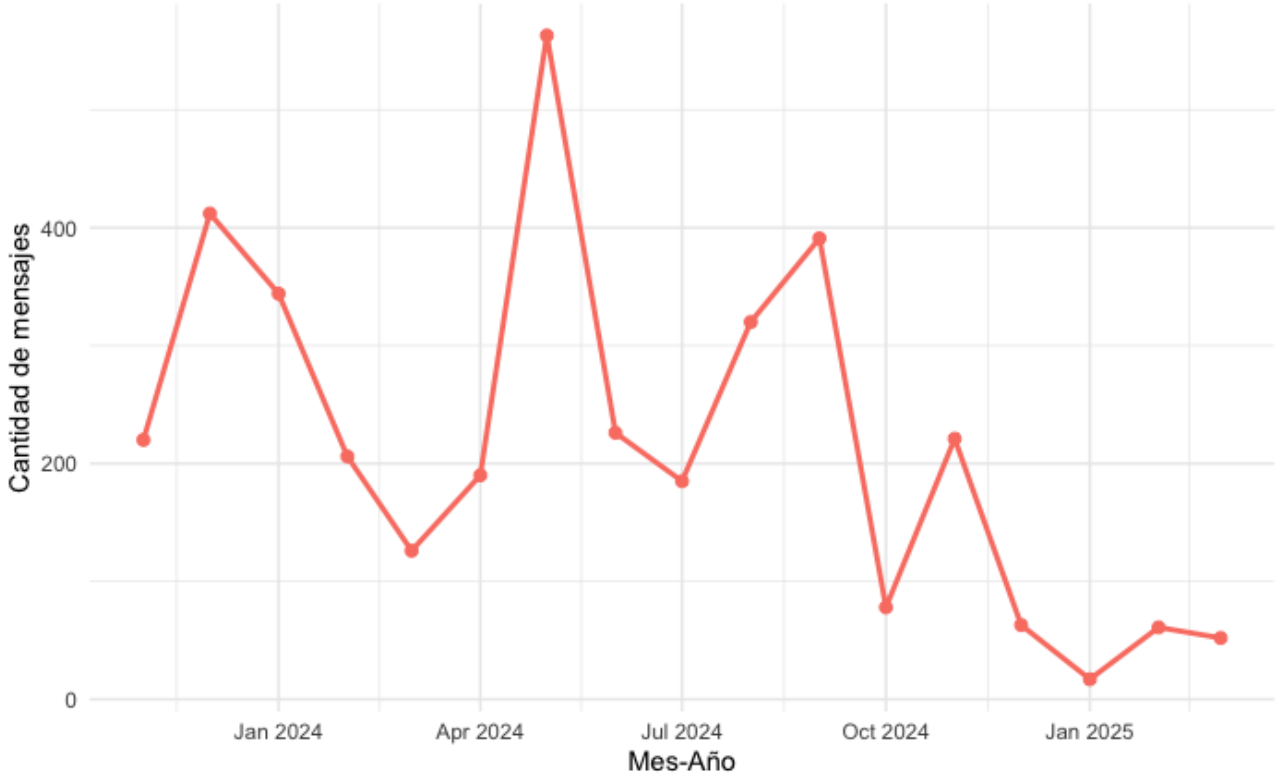
Fuente: Elaboración propia con la base de datos.

En la Tabla 2, se puede observar el rango de fechas que se está analizando, partiendo del 01 de noviembre del 2023 (fecha en que se creó el grupo) al 31 de marzo del 2025 (fecha corte para el análisis).

A su vez, se puede apreciar el rango de horas el cual abarca prácticamente todo el día, desde horas de la madrugada hasta altas horas de la noche. El mínimo corresponde a 01:12, mientras que el máximo sería las 12:59. Esto es causado porque la base de datos ha sido extraída de un dispositivo con un reloj de 12 horas, y por tanto, el valor mínimo posible será la 1:00 y el valor máximo posible son las 12:59, independientemente de la mitad del día. Dicho problema se puede solventar más adelante.

1.9. Distribución de Variables Cuantitativas

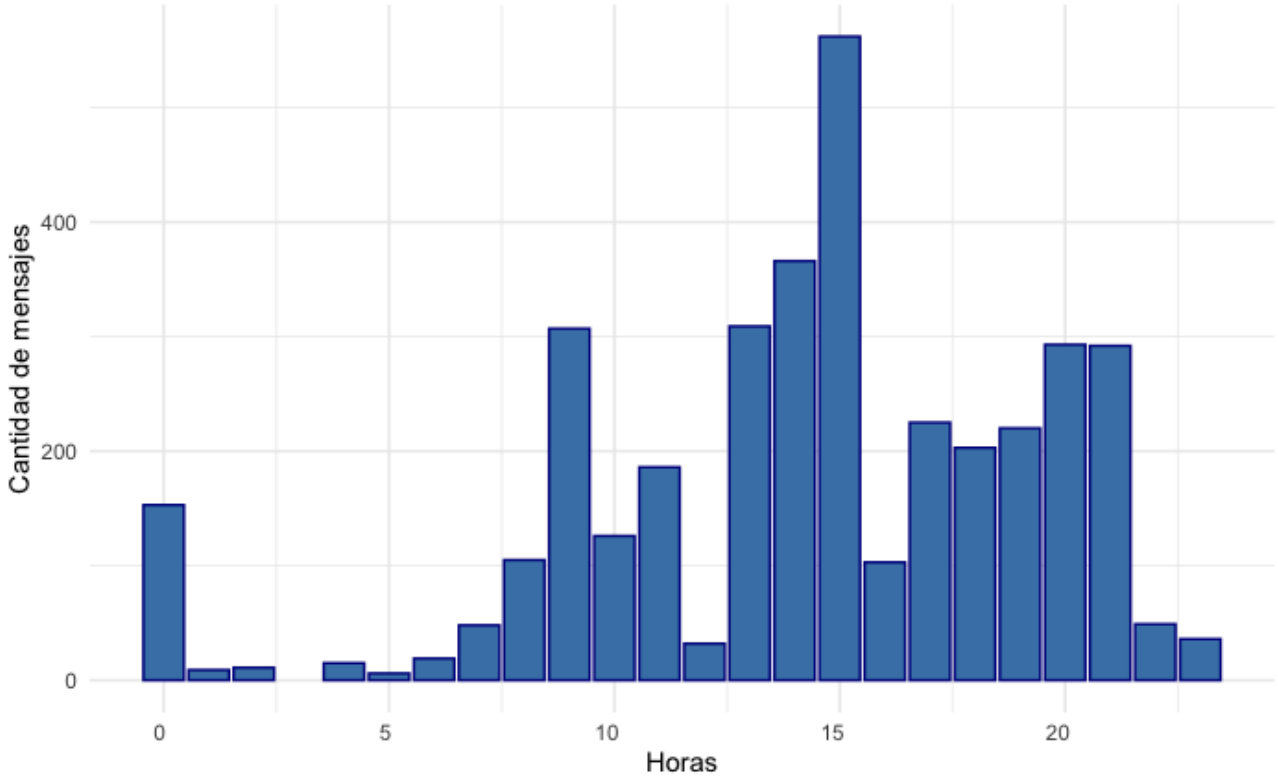
Figura 2: Cantidad de mensajes según el mes y año.



Fuente: Elaboración propia con la base de datos.

En la Figura 2 se puede ver la cantidad de mensajes por cada mes y año. A grandes rasgos, se observa una tendencia cada vez menor en la actividad del grupo, con un pico a finales del primer semestre del año 2024. También se observa poca actividad en horario de vacaciones, exceptuando el verano del 2024 donde los estudiantes llevaron un curso en conjunto. El mes más inactivo corresponde a enero del 2025.

Figura 3: Cantidad de mensajes por hora del día.

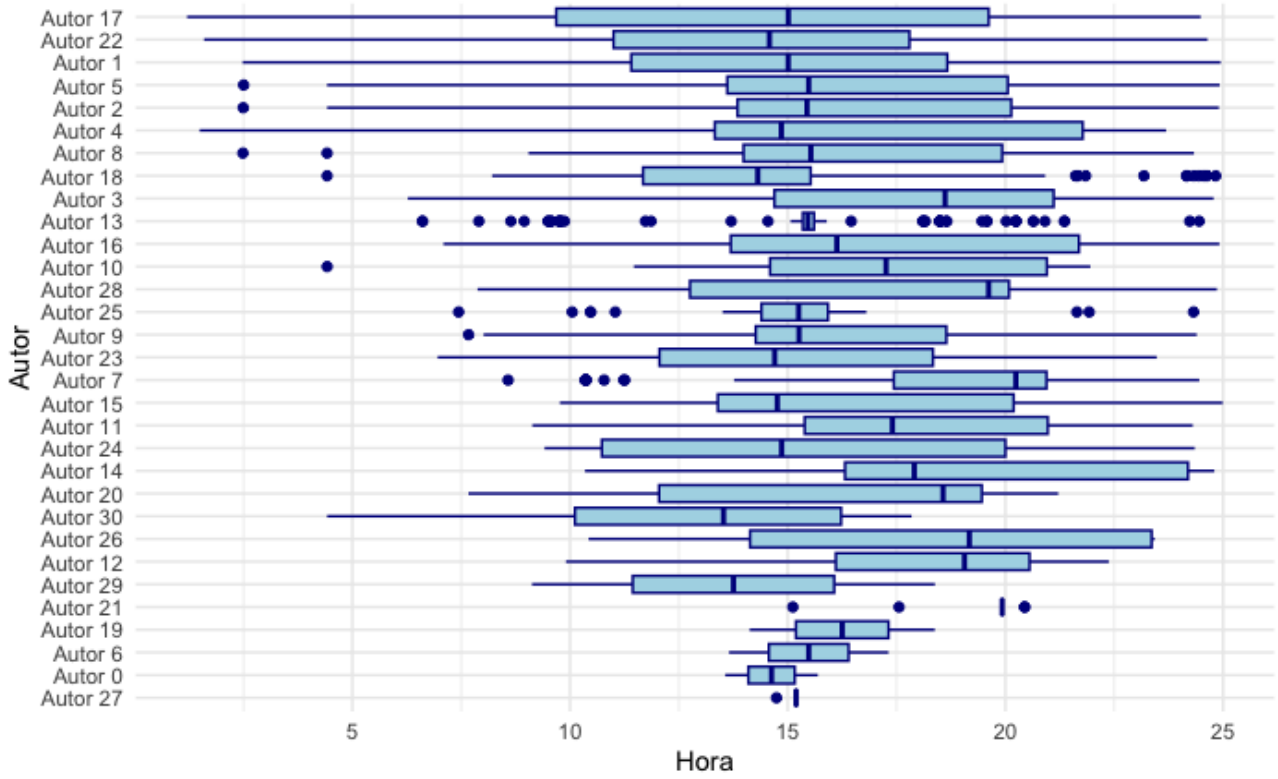


Fuente: Elaboración propia con la base de datos.

En la Figura 3 se muestran la cantidad de mensajes con respecto a la hora del día. En dicho gráfico se evidencia que no hay ningún mensaje entre las 3 y 4 de la mañana. A su vez, se nota la poca actividad en la hora de almuerzo, a las 12 del medio día. También se puede intuir débilmente que las horas de sueño promedio son entre la 1 y las 7 de la mañana, porque son las horas menos activas. También se observa que 12 horas después de la hora inactiva se tiene la hora más activa, alrededor de las 3 de la tarde.

1.10. Relación entre Variables

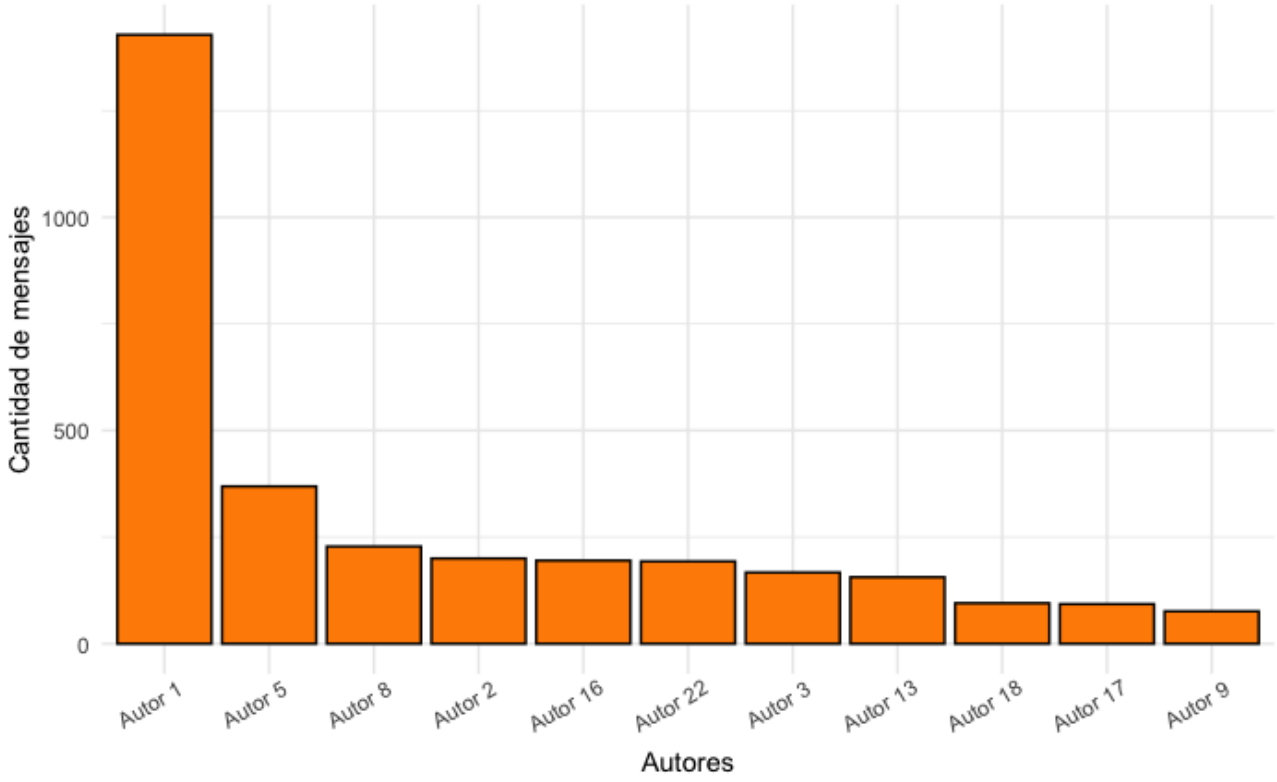
Figura 4: Distribución de las horas por autor.



Fuente: Elaboración propia con la base de datos.

En la Figura 4 se muestran diferentes estadísticas de las horas en las que escribe cada autor. En ella se pueden observar los rangos promedio en que cada autor escribe, junto a su máximo y mínimo. También resulta sencillo ver que si un autor ha enviado pocos mensajes a cierta hora fuera de lo común, entonces aparecen como puntos alejados del rango.

Figura 5: Mayor cantidad de mensajes por autor.



Fuente: Elaboración propia con la base de datos.

Para resumir las participaciones de los autores, se puede contabilizar la cantidad de mensajes de cada uno, como se muestra en la Figura 5. Al complementarlo con la figura Figura 4, se evidencia que un autor puede escribir mucho, pero no necesariamente tener un rango de horas amplio. También se puede notar la desproporción entre el autor que más escribe y el segundo. Este gráfico fue recortado para que muestre aquellos autores con más de 70 mensajes, pues las pocas participaciones no se pueden observar de manera clara ante los 1500 mensajes del primer autor.

1.11. Distribución de Variables Categóricas

Como ya se observó la distribución de los autores en varios aspectos, se puede realizar un gráfico más enfocado al objetivo del proyecto, lo que vendría siendo un mapa general de las palabras contenidas en los mensajes.

Figura 6: Mapa de palabras.

Fuente: Elaboración propia con la base de datos.

Es importante recalcar que en la Figura 6 no se han eliminado los mensaje no relevantes, como “sticker omitido”, que se repite muchas veces. A su vez, tampoco se han eliminado las preposiciones más comunes, las cuales aparecen repetidamente en el mapa de palabras, tales como: “a” o “que”. Por lo tanto, esta corresponde a una versión preliminar de las palabras que se van a analizar.

1.12. Valores Faltantes y Outliers

Originalmente, no se presentan valores faltantes en los datos, ya que estos provienen de una exportación directa de los chats de WhatsApp y, por tanto, se mantienen completos en términos de contenido. Sin embargo, esto no implica que el texto esté listo para su análisis. Debido a la naturaleza conversacional y espontánea de los mensajes, es necesario aplicar técnicas de limpieza y preprocesamiento, tales como la eliminación de palabras vacías, símbolos irrelevantes, mensajes automáticos como "sticker omitido", y otros elementos que no aportan información significativa para los fines analíticos del proyecto. Estas transformaciones son fundamentales para reducir el ruido y garantizar la calidad del análisis posterior, como el reconocimiento de patrones emocionales mediante algoritmos de clasificación, y serán explicadas a continuación.

1.13. Técnicas para tratar Valores perdidos y Outliers

- **Eliminación de palabras vacías:**

  De acuerdo al IBM, las palabras vacías "son palabras de aparición frecuente y sin contenido significativo para el proceso de recuperación de texto". (IBM, 2022) Usualmente, se considera que todas las palabras de función (en términos lingüísticos) son palabras vacías, tales como "y", "o" y "en".

  La eliminación de palabras vacías en los datos de texto puede llevarse a cabo de diversas formas, dependiendo del tipo, la procedencia y el objetivo de los datos. El método más común consiste en contrastar cada palabra del texto con una lista de palabras vacías preestablecida y filtrar las coincidencias. También es posible usar una lista estándar o elaborar una propia según su dominio y contexto. (Sawant et al., 2023)

  Aplicar esta técnica puede ser ventajoso si dichas palabras son insignificantes o redundantes

para el estudio de texto, y puede incrementar el desempeño y exactitud de los procedimientos y modelos. Igualmente, puede resultar beneficioso si los datos de texto son amplios, desorganizados o ruidosos, dado que puede disminuir la complejidad y la dimensión de los datos.

No obstante, no es una buena práctica el eliminar las palabras vacías si son relevantes o valiosas para su estudio de texto, dado que puede mantener el ritmo natural y gramatical del texto. Además, se debe evitar la eliminación de palabras vacías si los datos de texto son reducidos, organizados o ordenados, dado que pueden preservar la diversidad y la riqueza de la información.

- **Stemming y lematización:**

  Las técnicas de stemming y lematización son métodos de preprocesamiento de textos en el procesamiento del lenguaje natural. En particular, disminuyen las formas flexibles de las palabras de un conjunto de textos a una palabra raíz habitual o forma de diccionario, también denominada “lema” en la lingüística computacional. (Murel and Kavlakoglu, 2023)

  La derivación y la lematización resultan particularmente beneficiosas en sistemas de recuperación de datos como los motores de búsqueda, donde los usuarios pueden hacer una consulta con una única palabra (como meditar), pero anticipan resultados que empleen cualquier forma reducida de la palabra (como medita, meditación, etc.). El propósito de la stemming y la lematización también es optimizar el procesamiento de texto en algoritmos de aprendizaje automático. (Murel and Kavlakoglu, 2023)

  Sin embargo, para este proyecto, aplicar técnicas de stemming y lematización podría favorecer la identificación de patrones emocionales, ya que al reducir las palabras a sus raíces comunes, se disminuye la variabilidad del vocabulario y se agrupan términos que comparten un mismo significado base. Esto facilita el análisis posterior, mejora la representación semántica de los mensajes y permite que los modelos de clasificación trabajen con un conjunto de caracte-

rísticas más homogéneo y representativo del contenido emocional real de las conversaciones.

- **Expresiones regulares:**

  Las expresiones regulares, a menudo conocidas como RegEx (por sus siglas en inglés), son secuencias de caracteres muy utilizadas tanto en el ámbito computacional teórico como en la implementación de lenguajes de programación para ejecutar patrones de búsqueda en secuencias de texto. En otras palabras, son comodines empleados para efectuar búsquedas utilizando diversos caracteres especiales, ya sea de forma individual o en grupos de estos. (Oller Aznar, 2023)

  Este tipo de "atajos" nos facilitan la creación de secuencias que puedan acomodarse a diferentes textos, basándonos en un patrón común para dichas secuencias. De esta manera, podremos llevar a cabo una operación de búsqueda mediante un lenguaje de programación, escribiendo una única frase en vez de tener que redactar todas las combinaciones posibles. (Oller Aznar, 2023)

  Asimismo, en este proyecto, el uso de expresiones regulares (regex) puede ser una herramienta fundamental para mejorar la calidad del texto antes del análisis. A través de patrones definidos, se facilita la detección y eliminación de elementos irrelevantes o repetitivos, como mensajes automáticos del tipo "sticker omitido", "imagen omitida" o enlaces compartidos, que no aportan contenido semántico relevante para el análisis emocional.

1.14. Control de Versiones en GitHub

El repositorio se encuentra en el siguiente ***enlace***.

2. Bitácora 2

2.1. Marco metodológico

2.1.1. Preprocesamiento de los datos

En primer lugar, se le asigna una etiqueta a cada mensaje, esto con el fin de poder rastrear los mensajes más adelante. Este proceso se realiza de manera sencilla, asignándole un número consecutivo a cada mensaje.

Ahora bien, aunque la base de datos ya está limpia y ordenada, esta se encuentra con mensajes indeseados para el análisis de sentimientos, como lo pueden ser enlaces, imágenes, stickers, documentos, entre otros. Para solventar dicho problema, se procede a eliminar todas las observaciones que contengan estas palabras específicas, puesto que se revisa que el uso de ellas no vienen acompañados con texto que sí se le puede hacer tal análisis.

Para ello, se utiliza la función `grepl()` de R, la cual permite identificar si una cadena de texto contiene un patrón específico mediante expresiones regulares, técnica descrita anteriormente. Esta función devuelve un valor lógico (TRUE o FALSE) por cada observación, facilitando así la filtración de los mensajes que incluyen palabras clave como “sticker omitido”, “imagen omitida”, “video omitido” o enlaces web que inician con “http”. Posteriormente, se eliminan aquellas filas cuya columna de texto coincide con dichos patrones, asegurando que únicamente permanezcan mensajes susceptibles de ser analizados desde una perspectiva semántica.

Otra técnica común en el tratamiento de texto es la *tokenización*, que consiste en dividir un texto en unidades más pequeñas llamadas tokens, como palabras o caracteres. Este proceso es esencial en el Procesamiento del Lenguaje Natural (PLN) y en el aprendizaje automático, ya que permite transformar el lenguaje en fragmentos simples que las máquinas pueden analizar con mayor facilidad. Así, se

facilita el estudio de aspectos como la frecuencia de palabras o la identificación de sentimientos en los mensajes. (Awan, 2024)

Se considera también eliminar cualquier número entero, considerando que las menciones (@num-telefono) al tokenizarlas, se les elimina el signo de "@". Es importante destacar que el proceso de tokenización solo descompone las palabras del mensaje en un estilo "pivot-longer", donde se copian las demás columnas y se hacen observaciones extras por cada palabra en el mensaje. Esta tokenización se realiza de manera automática por la librería `tidytext`, la cual elimina algunos caracteres especiales y principalmente los emojis, que tampoco son relevantes para el análisis.

Por último se procede a eliminar cualquier palabra repetida más de 43 veces, se escogió este número puesto que cualquier palabra que sobrepase esta cantidad es altamente probable que no tenga ningún sentimiento asociado.

Figura 7: Mapa de palabras actualizado.

Fuente: Elaboración propia

En la Figura 7 se observa el mapa de palabras actualizado, dicho gráfico contiene las observaciones finales luego de eliminar los mensajes no relevantes explicados anterioemente. En este nuevo gráfico se puede ver una distribución más equitativa y a su vez se aprecian mejor las palabras que son más interpretativas, como algunos nombres o expresiones comunes del grupo.

2.1.2. Representación de los datos

En esta etapa, se utilizó el *NRC Word-Emotion Association Lexicon* para cuantificar las emociones presentes en los mensajes. Cada palabra extraída de los mensajes fue comparada contra el lexicón en español disponible mediante la función `get_nrc_sentiment()` de la librería `syuzhet`. Esta función asigna a cada palabra un conteo binario (presencia/ausencia) respecto a ocho emociones básicas (ira, anticipación, disgusto, miedo, alegría, tristeza, sorpresa y confianza) y dos polaridades (positivo y negativo). Durante la implementación, para cada mensaje se sumó el número de palabras asociadas a cada emoción, generando así un perfil emocional agregado para cada mensaje. (Mohammad and Turney, 2013)

En este proyecto, se utilizó una estrategia de conteo simple, en la cual cada aparición de una palabra asociada incrementa en uno la emoción correspondiente del mensaje, sin ponderar por intensidad. Esta simplificación permite capturar de manera eficiente las tendencias emocionales generales.

Formalmente, sea un mensaje

$$m_j = (w_1, w_2, \ldots, w_n)$$

donde cada w_i es una palabra, y sea $\mathcal{E}$ el conjunto de emociones consideradas, con $|\mathcal{E}| = p$. Definimos, para cada emoción $e \in \mathcal{E}$, el conteo asociado al mensaje m_j como:

$$x_e(m_j) = \sum_{i=1}^{n} \mathbf{1}\{w_i \in \mathcal{W}_e\}$$

donde $\mathbf{1}\{\cdot\}$ es la función indicadora que toma el valor 1 si w_i pertenece al conjunto de palabras asociadas a la emoción e en el lexicón, y 0 en caso contrario.

Así, cada mensaje m_j queda representado por un vector emocional:

$$X(m_j) = (x_{e_1}(m_j), x_{e_2}(m_j), \ldots, x_{e_p}(m_j)) \in \mathbb{R}^p$$

donde cada componente $x_{e_k}(m_j)$ corresponde al conteo de palabras asociadas a la emoción e_k en el mensaje m_j. Este vector es posteriormente utilizado para análisis de agrupamiento y clasificación, considerando el espacio emocional generado por las frecuencias relativas de cada emoción.

2.1.3. Algoritmos de clasificación

En este proyecto se aplican dos variantes del algoritmo de clasificación no supervisado, siendo el K-Medias tradicional (K-Means), empleando la media aritmética como criterio de actualización de centroides (Trejos et al., 2021). También se implementa el K-Medianas (una variante basada en la mediana), utilizando la mediana en lugar de la media para la actualización de los centroides, con el objetivo de incrementar la resistencia frente a valores atípicos (Aggarwal, 2015). Ambos métodos son implementados sobre representaciones numéricas de los mensajes de texto, codificados en función de su contenido emocional.

El objetivo de los métodos es particionar el conjunto de datos $X = \{x_1, x_2, \ldots, x_n\}$ en k clusters $C_1, C_2, \ldots, C_k$ de manera que se minimice la suma de las distancias cuadradas de cada observación a su centroide:

$$E = \sum_{j=1}^{k} \sum_{x_i \in C_j} d^2(x_i, \mu_j)$$

donde $d(x_i, \mu_j)$ representa la distancia euclidiana entre la observación x_i y su centroide μ_j.

El cálculo de los centroides se da dependiendo de la estrategia de actualización consideradas anteriormente:

- **K-Medias tradicional**: El centroide μ_j se calcula como el promedio de las observaciones asignadas a C_j (Trejos et al., 2021):

$$\mu_j = \frac{1}{|C_j|} \sum_{x_i \in C_j} x_i$$

- **K-Medianas**: El centroide $\tilde{\mu}_j$ se calcula como la mediana componente a componente de las observaciones en C_j (Aggarwal, 2015):

$$\tilde{\mu}_j = (\text{mediana}(x_{i1}), \ldots, \text{mediana}(x_{ip})) \quad \forall x_i \in C_j$$

Para ambos algoritmos se desarrolla el procedimiento dado por:

1. Se seleccionan aleatoriamente k observaciones del conjunto de datos como centroides iniciales.
2. Cada observación x_i se asigna al cluster cuyo centroide esté más cercano, utilizando la distancia euclidiana.
3. Se recalculan los centroides de acuerdo al método escogido (media o mediana).
4. Se repiten los pasos 2 y 3 hasta que no haya cambios en las asignaciones o se alcance un máximo de iteraciones.

Este procedimiento sigue las formulaciones presentadas en Trejos et al. (2021) para el caso de medias y en Aggarwal (2015) para el caso de medianas.

2.2. Ajuste del modelo

2.2.1. Definición de los modelos

Se trabajará con los siguientes modelos, todos partiendo de la misma base de datos:

- Modelo 1: Datos originales sin ninguna modificación (completo).
- Modelo 2: Datos sin las palabras vacías y sin las palabras repetidas.*
- Modelo 3: Modelo 1 pero eliminando todo mensaje cuyos sentimientos sea 0.
- Modelo 4: Modelo 2 pero eliminando todo mensaje cuyos sentimientos sea 0 (simple).

*Para el modelo 2 y 4 su proceso se detalla en la metodología.

2.2.2. Primer intento de ajuste del modelo

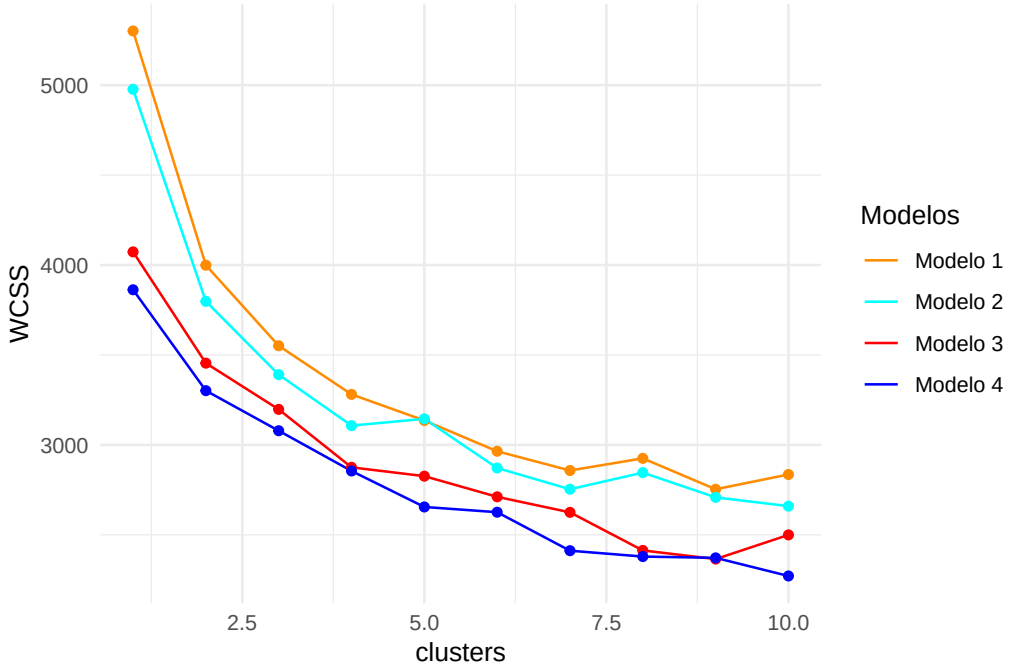
Para aplicar K-medias a estos modelos, se implementa de las dos formas redactadas anteriormente, usando el promedio para encontrar el centroide en cada iteración, o usando la mediana y observar qué observación está más cerca del centroide para asignarlo como tal. Ambos métodos llevan a resultados diferentes, además de que tomar el promedio es mucho más eficiente. La idea detrás de asignar en cada iteración una observación como centroide le da un resultado más interpretativo, porque a un centroide con un sentimiento promedio puede ser abstracto.

Al mismo tiempo, se puede calcular la métrica *WCSS*, la cual mide la separación de los centroides con cada una de sus observaciones asignadas en el clúster respectivo. En este caso, se puede calcular tanto para la media como la mediana, pero recordando que se usa una iteración arbitraria (la selección de clústeres en cada paso puede no ser la más óptima).

2.2.3. Análisis y Resultados

Para comenzar, se le aplica el algoritmo de K-Medias a todos los modelos descritos anteriormente y se utiliza la métrica *WCSS* para comparar los resultados:

Figura 8: Métricas WCSS con K-Medias.



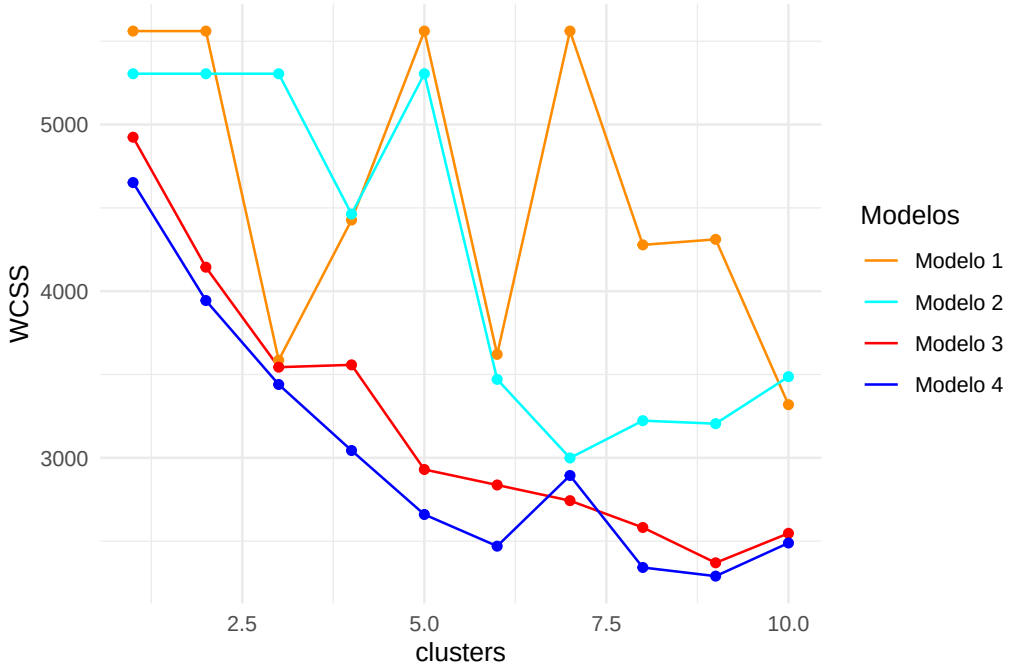
Fuente: Elaboración propia.

Se puede observar en el gráfico de la Figura 8 que el modelo 4 resulta con la menor métrica, y esto significa que los clústeres se encuentran relativamente más agrupados, o más cerca de sus centroides. Además, al tomar este modelo, se observa un "codo" en 5 clústeres, por lo que se decide tomar esta cantidad para analizar los clústeres más repetidos, como se realizará más adelante.

Por otro lado, al implementar el algoritmo de K-Medianas, se tiene un problema a la hora de analizar los primeros dos modelos, ya que hay una gran cantidad de mensajes sin sentimientos y al tomar la mediana habrán muchos centroides que se mantendrán en cero. Esta fue la principal razón

para tomar en cuenta los modelos 3 y 4, como se verá en el gráfico a continuación:

Figura 9: Métricas WCSS con K-Medianas.



Fuente: Elaboración propia.

Como se puede observar en el gráfico de la Figura 9, los primeros dos modelos parecen mejorar arbitrariamente con la cantidad de clústeres que se le agregan, hasta a veces empeorar, ya que al tener muchas observaciones con ceros sentimientos, la asignación de los centroides no resulta óptima, como ya fue descrito anteriormente. En cambio, por los dos últimos modelos se observa una tendencia similar a la metodología anterior, el K-Medias tradicional. Para ser consistentes, se decide tomar también 5 clústeres como la cantidad de clústeres a analizar, ya que en el modelo 4 no se observa un "codo" como tal.

Por otro lado, a la hora de analizar los clústeres, se decide partir para un mejor análisis desde la cantidad descrita anteriormente y usando el modelo 4, ya que se debe enfocar en un resultado más

interpretativo y contundente. Esto se realiza etiquetando cada mensaje con un `id` y luego rastreando los centroides con sus respectivos `id`. Se ordenan los centroides después de correr bastantes veces la función K-Medianas con los atributos descritos y se hace un resumen para ver qué set de centroides se repite con más frecuencia. A continuación en la tabla se encuentran estos resultados:

Tabla 3: Set de centroides con más repeticiones.

Set de centroides	**Valor**
2-41-98-142-373	9
40-98-142-216-373	9
40-41-98-142-216	8
2-41-98-142-2062	7
2-41-98-216-2062	7
2-41-98-216-373	7
40-98-216-254-373	7
⋮	⋮

Fuente: Elaboración propia.

Con 5 centroides se dispersan más las posibilidades de la escogencia de los centroides, así que su repetición es bastante improbable. De las 1000 ejecuciones de la función, estas observaciones de la Tabla 3 son las más repetidas. Se puede hacer el mismo análisis pero para cada centroide individual, como se evidencia en la Tabla 4. En esta tabla se pueden ver los centroides de la Tabla 3 repitiéndose la mayor cantidad de veces.

Tabla 4: Centroides con más repeticiones.

Centroide	**98**	**40**	**2**	**373**	**142**	**15**	**216**	**2062**	**41**	**99**	**43**	**355**
Frecuencia	861	539	494	342	317	241	229	214	205	181	113	106

Fuente: Elaboración propia.

Tabla 5: Distribución de emociones en el set 2-41-98-142-373.

Emoción	**Cluster 1**	**Cluster 2**	**Cluster 3**	**Cluster 4**	**Cluster 5**
anger	23	104	0	0	12
anticipation	24	29	4	57	132
disgust	27	99	5	1	6
fear	70	107	3	10	35
joy	0	9	0	0	152
sadness	98	97	1	5	29
surprise	40	24	5	1	62
trust	14	18	110	1	218
negative	260	226	0	1	45
positive	30	36	55	13	358
# Observaciones	208	99	108	55	170

Fuente: Elaboración propia.

Ahora bien, tomando el set de clústeres más repetido (2-41-98-142-373) se hace la extracción de sentimientos para cada uno de los grupos. Se nota que en el primero y segundo hay una predominancia negativa, aunque en el segundo los sentimientos de `anger, disgust, fear` son mayores. En el

tercero hay más presencia del sentimiento de confianza, mientras que el cuarto es el más neutral. El último grupo tiene más valores de `anticipation, joy, trust, positive`, considerándose el más positivo y feliz.

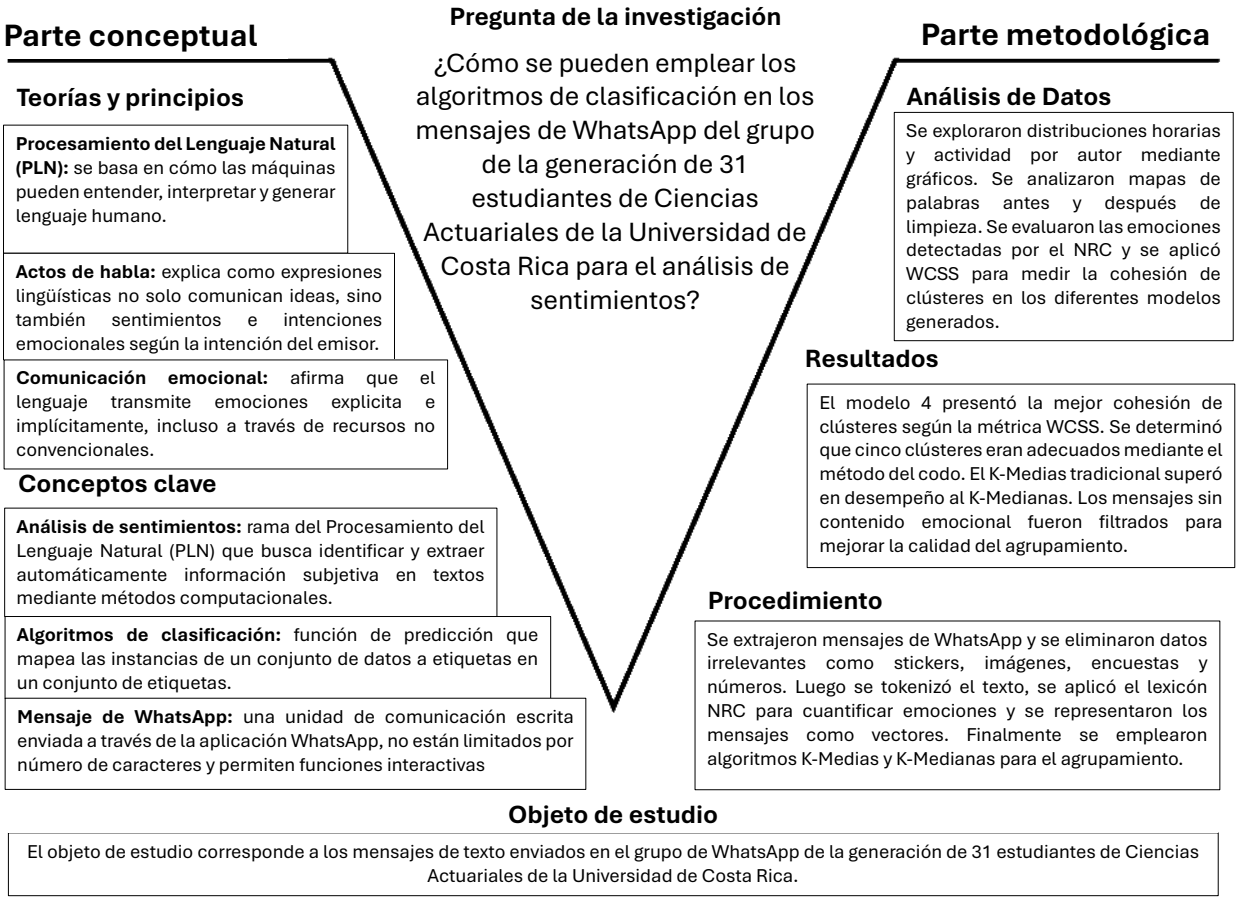
2.2.4. Respuestas preliminares a la pregunta de investigación

Recordando la pregunta de investigación: *¿Cómo se pueden emplear los algoritmos de clasificación en los mensajes de WhatsApp del grupo de la generación de 31 estudiantes de Ciencias Actuariales de la Universidad de Costa Rica para el análisis de sentimientos?*

De manera general, en esta bitácora se lograron emplear los métodos K-Medias y K-Medianas para poder analizar cuál de los 4 modelos propuestos era el más óptimo de analizar. Después de realizar la tokenización y extracción de los sentimientos en cada mensaje, al agrupar los mensajes se pueden realizar bastantes análisis subsecuentes con los centroides y los grupos dados. La otra técnica propuesta es implementar los mismos métodos, pero en vez de agrupar por mensajes, agrupar por autores y sus sentimientos. Hay distintas formas de emplear el análisis de sentimientos en una base de datos como la dada, pero de momento se le va a dar énfasis a la presentada en esta bitácora debido a los resultados positivos obtenidos.

2.3. Uve de Gowin actualizada

Figura 10: Diagrama Uve de Gowin actualizada.



Fuente: Elaboración propia.

2.4. Anotaciones del profesor y de los compañeros.

Se llevó a cabo una corrección minuciosa sobre todas las observaciones realizadas por el profesor de la Bitácora 1, de las cuales se destacan las siguientes:

- Mayor contextualización de lo realizado en el proyecto anterior.
- Reformulación de la pregunta de investigación, dándole un enfoque más acorde a lo que se está realizando en el proyecto.
- Censura a los nombres de los autores de los mensajes, esto con el fin de mantener la integridad de los mismos.
- Búsqueda de técnicas respaldadas con bibliografía para tratar los mensajes poco relevantes para el análisis.

A su vez, se realizaron correcciones de formato que fueron las principales observaciones realizadas por los compañeros, y se tomaron en consideración las propuestas algorítmicas que podrían facilitar el trabajo.

2.5. Control de Versiones en GitHub

El repositorio se encuentra en el siguiente ***enlace***.

3. Bitácora 3

3.1. Marco Metodológico

Anteriormente se detalló el proceso de análisis exploratorio, el preprocesamiento de los mensajes y su clasificación utilizando los algoritmos de *k-means* y *k-medianas*, los cuales ofrecieron una primera aproximación útil para agrupar los textos según su contenido. No obstante, al tratarse de métodos ampliamente estudiados durante el curso, en esta sección se propone un enfoque alternativo que permita ampliar la perspectiva metodológica del proyecto.

3.1.1. Mapas Auto-Organizativos

Los Mapas Auto-Organizativos (SOM, por sus siglas en inglés[1]) son una técnica de aprendizaje no supervisado propuesta por Teuvo Kohonen en los años ochenta (Cottrell et al., 2018). Este modelo combina capacidades de agrupamiento con una poderosa herramienta de visualización, lo que lo hace especialmente adecuado para representar y explorar conjuntos de datos complejos y de alta dimensión, como lo son los mensajes transformados en vectores numéricos.

A diferencia de los métodos que sólo generan particiones en los datos, como *k-means*, los SOM proyectan los datos a un espacio bidimensional preservando las relaciones topológicas del espacio original. Es decir, instancias similares en el espacio de entrada tienden a ubicarse en nodos vecinos dentro del mapa, lo que facilita la interpretación de patrones o estructuras subyacentes (Cottrell et al., 2018; Wehrens and Buydens, 2007).

Este enfoque resulta particularmente útil en contextos donde no se dispone de etiquetas previas, como en el análisis de sentimientos de conversaciones escritas. Al aplicar SOM a mensajes vectorizados, se pueden detectar regiones del mapa con patrones similares de contenido o tono emocional,

[1] *Self-Organizing Maps*

permitiendo así una exploración más rica del comportamiento del grupo analizado.

Originalmente, el algoritmo SOM fue definido para datos representados por vectores numéricos pertenecientes a un subconjunto $X \subset \mathbb{R}^p$, es decir, datos que viven en un espacio euclidiano de dimensión finita. Sin embargo, desde el punto de vista matemático se pueden distinguir dos escenarios: el continuo, donde los datos provienen de una distribución de probabilidad con densidad f; y el discreto, que corresponde al caso práctico habitual en el que se dispone de un conjunto finito de observaciones $\{x_1, \ldots, x_n\}$.

Formalmente, un SOM consiste en una grilla de unidades (o neuronas) organizadas en una estructura topológica regular, usualmente rectangular o hexagonal, donde cada unidad i está asociada a un vector de pesos $\mathbf{w}_i \in \mathbb{R}^d$. Dado un vector de entrada $\mathbf{x} \in \mathbb{R}^d$, se identifica la unidad ganadora, o *Best Matching Unit* (BMU), como la más cercana a $\mathbf{x}$ según una métrica, típicamente euclidiana:

$$i^* = \arg\min_i \|\mathbf{x} - \mathbf{w}_i\|$$

Una vez localizada la BMU, se actualizan los pesos de esa unidad y de sus vecinas mediante la regla:

$$\mathbf{w}_i(t+1) = \mathbf{w}_i(t) + \alpha(t) \cdot h_{i,i^*}(t) \cdot (\mathbf{x} - \mathbf{w}_i(t))$$

donde:

- $\alpha(t)$: es la tasa de aprendizaje, una función decreciente en el tiempo,
- $h_{i,i^*}(t)$: es la función de vecindad que disminuye con la distancia entre la unidad i y la unidad ganadora i^*, y también con el tiempo.

La vecindad suele definirse mediante una función gaussiana centrada en i^*:

$$h_{i,i^*}(t) = \exp\left(-\frac{\text{dist}^2(i, i^*)}{2\sigma(t)^2}\right)$$

donde $\sigma(t)$ controla el ancho de la vecindad y se reduce gradualmente durante el entrenamiento, y esta debe satisfacer las siguientes propiedades:

- h es simétrica y $h_{i,i} = 1$,
- h_{i,i^*} depende únicamente de la distancia $\text{dist}(k, l)$ en la grilla y desciende cuando incrementa la distancia.

Este proceso se repite durante múltiples iteraciones, permitiendo que el mapa se "organice" a sí mismo, es decir, que los vectores de pesos converjan hacia representaciones estructuradas del espacio de datos. El resultado final es una representación en la que la disposición espacial de las unidades refleja relaciones de similitud entre los datos originales (Cottrell et al., 2018; Wehrens and Buydens, 2007).

Desde una perspectiva estadística, los SOM pueden considerarse una extensión espacialmente restringida del algoritmo *k-means*, donde además de minimizar la distancia a un centroide, se impone una estructura topológica que preserva la continuidad del espacio (Cottrell et al., 2018). Esto implica que los SOM no solo realizan una cuantización del espacio de datos (como lo hace *k-means*), sino que también mantienen una organización que facilita la visualización y exploración de relaciones semánticas o sentimentales latentes.

En la implementación de este proyecto se utilizó la versión estándar del algoritmo, conocida como SOM en línea, tal como está disponible en el paquete `kohonen` de R. Este paquete fue desarrollado con un fuerte énfasis en la visualización, y proporciona funciones sencillas y versátiles tanto para mapas auto-organizativos clásicos como para variantes más avanzadas. Entre sus funciones principales se encuentran `som` para modelos no supervisados, `xyf` para mapas supervisados (cuando se cuenta con

una variable de clase), y `supersom` para trabajar con múltiples capas de información (Wehrens and Buydens, 2007).

La configuración del SOM en este proyecto se definió con base en criterios teóricos y consideraciones prácticas. Se utilizó una grilla pequeña de nueve unidades, adecuada para datos de tamaño moderado y con fines exploratorios, siguiendo recomendaciones que sugieren mantener un balance entre resolución y simplicidad interpretativa (Cottrell et al., 2018). La topología elegida fue hexagonal, debido a su mejor preservación de relaciones topológicas y su conectividad más uniforme entre unidades (Wehrens and Buydens, 2007). Además, se utilizó un número de iteraciones suficiente para asegurar la convergencia del modelo, y los datos fueron escalados previamente, dado que el SOM es sensible a la magnitud de las variables.

Uno de los principales aportes del paquete `kohonen` es la variedad de herramientas visuales que facilita para interpretar y evaluar el mapa entrenado. Entre ellas, se destaca el gráfico de tipo `codes`, que representa los vectores de referencia, aprendidos por cada unidad del mapa. Esta visualización proyecta, en cada nodo, un conjunto de círculos que resumen el perfil promedio de las observaciones asignadas a esa unidad, permitiendo identificar patrones en la distribución de las variables originales.

En este proyecto, estos vectores capturan la intensidad relativa de distintas emociones detectadas en los mensajes, lo que ayuda a interpretar qué tipo de carga emocional caracteriza a cada región del mapa. De esta forma es posible identificar zonas del SOM asociadas a emociones como alegría, tristeza o enojo, dependiendo del contenido de los mensajes agrupados.

3.1.2. DBSCAN

El algoritmo DBSCAN (Density-Based Spatial Clustering of Applications with Noise) es una técnica de aprendizaje no supervisado diseñada para identificar agrupamientos en conjuntos de datos complejos, especialmente cuando los clústeres tienen formas arbitrarias y no están bien separados.

A diferencia de métodos como *k-means*, DBSCAN no requiere especificar la cantidad de clústeres a priori, lo que resulta útil en contextos exploratorios como el análisis de sentimientos, donde no siempre se conoce de antemano cuántos grupos emocionales podrían emerger en los datos (Hahsler et al., 2019; Zhou et al., 2020).

La idea principal detrás de DBSCAN es que un clúster puede definirse como una región densa de puntos, separada de otras regiones menos densas o vacías. Para ello, el algoritmo se apoya en dos hiperparámetros fundamentales:

- ϵ: el radio de vecindad que determina hasta qué distancia se consideran çercanos"dos puntos,
- MinPts: la cantidad mínima de puntos dentro del vecindario ϵ que se requiere para considerar que hay densidad suficiente.

Con base en estos parámetros, DBSCAN clasifica los puntos en tres categorías:

- Puntos núcleo: tienen al menos MinPts vecinos en su radio ϵ.
- Puntos frontera: están dentro del vecindario de un núcleo, pero no tienen suficientes vecinos para ser núcleo por sí solos.
- Ruido: puntos que no pertenecen a ningún clúster, quedan aislados.

Matemáticamente, el vecindario ϵ viene dado por:

$$N_\epsilon(p) = \{q \in D : dist(p, q) \leq \epsilon\}$$

Donde D es el conjunto total de datos y la distancia suele calcularse con la métrica euclidiana. Un clúster se forma cuando varios puntos núcleo están conectados por densidad, es decir, cada uno puede alcanzarse a partir del anterior por una cadena de vecinos densamente conectados (Ester et al., 1996).

Formalmente:

$$N_\epsilon(p) > \mathrm{MinPts}$$

Una de las mayores fortalezas de DBSCAN es su capacidad para detectar clústeres de forma arbitraria, incluso si presentan bordes irregulares o no convexos. Además, el algoritmo es robusto frente a valores atípicos, ya que los puntos ruidosos se detectan y separan automáticamente sin requerir intervención adicional. Sin embargo, la elección adecuada de ϵ y MinPts es crítica, pues si se elige un ϵ demasiado pequeño, se generarán muchos clústeres pequeños y por ende, mucho ruido; mientras que si es demasiado grande, clústeres distintos pueden unirse indebidamente (Zhou et al., 2020).

Desde un punto de vista teórico, DBSCAN puede entenderse como un proceso de exploración local de densidad. A partir de un punto núcleo inicial, se expande el clúster agregando todos los puntos densa y transitivamente alcanzables. Este enfoque evita supuestos globales sobre la forma de los datos, lo cual lo vuelve ideal para aplicaciones como la segmentación de mensajes, donde los patrones emocionales o de estilo no necesariamente siguen estructuras geométricas simples (Rehman et al., 2014).

En este proyecto, DBSCAN se aplicó directamente a los vectores numéricos derivados de los mensajes de WhatsApp, previamente escalados. Para seleccionar los parámetros, se toma $\epsilon = 1$, esto debido a que como los datos están escalados, una distancia de 1 representa una vecindad razonable. Mientras que la elección de $\mathrm{MinPts} = 20$ es debido a que por recomendaciones empíricas, este debería ser igual a la dimensión del espacio + 1, y en muchos casos se recomienda un valor entre 10 y 50 para evitar clústeres espurios (Ester et al., 1996; Hahsler et al., 2019).

3.1.3. Métodos secundarios

Además de los modelos principales presentados en detalle, se exploraron otras metodologías de clasificación no supervisada con el objetivo de complementar los hallazgos, validar la consistencia de los agrupamientos observados y contrastar su comportamiento bajo distintos supuestos estructurales. Si bien estos modelos secundarios no constituyen el eje central del análisis, su inclusión responde a una intención comparativa y exploratoria. A continuación, se resumen las características principales de estas metodologías.

- **K-Means y K-Medians**

  Los algoritmos K-Means y K-Medians son métodos de clasificación no supervisada que buscan particionar un conjunto de datos $X = \{x_1, x_2, \ldots, x_n\}$ en k clústeres disjuntos $C_1, \ldots, C_k$, minimizando una función objetivo basada en la distancia entre cada punto y su centroide. En ambos casos, la asignación se realiza mediante distancia euclidiana, pero difieren en cómo se actualizan los centroides:

$$E = \sum_{j=1}^{k} \sum_{x_i \in C_j} d^2(x_i, \mu_j)$$

  - En K-Means, el centroide μ_j se calcula como la media de las observaciones en el clúster:

$$\mu_j = \frac{1}{|C_j|} \sum_{x_i \in C_j} x_i$$

  - En K-Medians, el centroide $\tilde{\mu}_j$ se calcula como la mediana componente a componente:

$$\tilde{\mu}_j = (\text{mediana}(x_{i1}), \ldots, \text{mediana}(x_{ip}))$$

Ambos métodos siguen un procedimiento iterativo: inicialización aleatoria de centroides, asignación de puntos al centroide más cercano, actualización de centroides y repetición hasta convergencia. K-Medians ofrece mayor robustez frente a valores atípicos (Trejos et al., 2021; Aggarwal, 2015).

- **Clustering Jerárquico**

  El algoritmo de clustering jerárquico fue uno de los métodos vistos durante el curso, y su inclusión en este proyecto permite comparar nuevos enfoques con técnicas ya conocidas y estudiadas. A diferencia de otros algoritmos que requieren definir a priori el número de clústeres, el enfoque jerárquico construye una estructura en forma de árbol que permite explorar agrupamientos a diferentes niveles de granularidad (Anónimo, sf).

  El clustering jerárquico aglomerativo es el más utilizado en la práctica, este comienza considerando cada observación como un clúster individual y, paso a paso, va fusionando los más similares hasta que todos los datos forman un solo grupo. Esta fusión se realiza en función de una medida de distancia y un criterio de enlace. Algunos de los criterios más comunes son:

  - **Enlace simple**: utiliza la distancia mínima entre dos puntos de clústeres distintos.
  - **Enlace completo**: utiliza la distancia máxima entre elementos de los dos clústeres.
  - **Enlace promedio**: calcula el promedio de todas las distancias posibles entre puntos de ambos clústeres.

  Cada una de estas estrategias genera una jerarquía distinta de agrupamientos y afecta la forma final del dendrograma. En este proyecto se utilizó el método de enlace `ward.D2`, una variante moderna del método de Ward que minimiza el incremento de la varianza intra-clúster en cada fusión. Esta estrategia tiende a formar agrupamientos compactos y bien definidos, lo cual

es especialmente adecuado para datos previamente escalados, como los vectores derivados de mensajes de texto (Anónimo, sf). Además, se seleccionó un número de clústeres igual a cinco mediante un corte controlado del dendrograma, con el fin de facilitar la comparación con otros métodos empleados en el análisis.

- **Modelos de Mezcla Gaussiana (GMM)**

  Los GMM son modelos probabilísticos que representan la distribución de los datos como una combinación de varias distribuciones normales multivariadas. Cada componente de la mezcla corresponde a un clúster latente, y se asume que los datos se generan a partir de una de estas distribuciones con cierta probabilidad. La función de densidad del modelo se expresa como:

$$p(\mathbf{x}) = \sum_{k=1}^{K} \pi_k \, \mathcal{N}(\mathbf{x} \mid \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$$

  donde π_k son los pesos de mezcla, con $\sum_{k=1}^{K} \pi_k = 1$, y $\mathcal{N}(\cdot)$ representa la densidad normal multivariada con media $\boldsymbol{\mu}_k$ y matriz de covarianza $\boldsymbol{\Sigma}_k$. El ajuste de los parámetros del modelo se realiza mediante el algoritmo de *Expectation-Maximization* (EM), que estima iterativamente las probabilidades de pertenencia y los parámetros de cada componente de la mezcla (Bishop, 2006).

- **Agrupamiento Espectral**

  Es una técnica de particionado de datos que utiliza los valores y vectores propios de una matriz de similitud construida a partir de los datos para realizar la agrupación. A diferencia de métodos clásicos como k-means, que operan directamente sobre las coordenadas originales, el agrupamiento espectral transforma los datos al espacio de los autovectores, donde la estructura del grafo de similitud se vuelve más evidente.

Sea $\mathbf{W}$ una matriz de afinidad entre los puntos, se construye una matriz Laplaciana $\mathbf{L}$, a partir de la cual se obtienen los k primeros vectores propios. Estos vectores se utilizan como nuevas representaciones de los datos, y sobre ellos se aplica un algoritmo de agrupamiento como k-means. Este método permite detectar estructuras no convexas y es especialmente útil cuando los datos forman grupos conectados de manera no lineal (von Luxburg, 2007).

- **Modelos de Mezcla No Gaussianos**

  Además de las mezclas gaussianas, es posible construir modelos de mezcla utilizando componentes que no siguen una distribución normal. Un caso común son las mezclas finitas con distribuciones t-Student, las cuales ofrecen mayor robustez frente a distribuciones con colas pesadas.

  Sea $\mathbf{x} \in \mathbb{R}^d$, la densidad de una mezcla de distribuciones t se expresa como:

$$p(\mathbf{x}) = \sum_{k=1}^{K} \pi_k \, t_\nu(\mathbf{x} \mid \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$$

  donde:

  - π_k son los pesos de mezcla, con $\sum_{k=1}^{K} \pi_k = 1$,
  - $t_\nu(\cdot \mid \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$ es la densidad de la distribución t multivariada con ν grados de libertad, media $\boldsymbol{\mu}_k$ y matriz de escala $\boldsymbol{\Sigma}_k$.

  Al igual que en las mezclas gaussianas, los parámetros se estiman mediante el algoritmo EM, con una E-step modificada que incorpora una variable latente adicional relacionada con la escala local de cada observación. Estos modelos permiten representar mejor datos contaminados (McLachlan and Peel, 2000).

- **Mean Shift Clustering**

  Este algoritmo es un método no paramétrico de agrupamiento, detecta automáticamente los modos, que se definen como máximos locales de una función de densidad en una distribución de datos, sin necesidad de especificar previamente el número de clústeres. Para cada punto $\mathbf{x} \in \mathbb{R}^d$, se calcula una dirección de ascenso hacia regiones de mayor densidad mediante el vector de *mean shift*:

$$\mathbf{m}(\mathbf{x}) = \frac{\sum_{i=1}^{n} K_h(\mathbf{x}_i - \mathbf{x})\mathbf{x}_i}{\sum_{i=1}^{n} K_h(\mathbf{x}_i - \mathbf{x})} - \mathbf{x}$$

  donde K_h es una función kernel (por ejemplo, gaussiana) con ancho de banda h, y $\mathbf{x}_i$ son los datos. Cada punto se actualiza iterativamente en la dirección de $\mathbf{m}(\mathbf{x})$ hasta converger a un modo de la densidad. Los puntos que convergen al mismo modo se agrupan en el mismo clúster (Comaniciu and Meer, 2002).

3.1.4. Comparación de Modelos

Con el objetivo de evaluar la calidad de las distintas segmentaciones generadas por los algoritmos de agrupamiento aplicados en este proyecto, se emplearon varias métricas de validación interna. Estas métricas permiten cuantificar la coherencia y separación de los clústeres sin necesidad de etiquetas externas, lo cual es especialmente útil en contextos no supervisados como el análisis de mensajes. A continuación, se describen las principales medidas utilizadas.

- **Coeficiente de Silhouette**

  Este coeficiente fue introducido por Peter J. Rousseeuw en su artículo titulado *"Silhouettes: a Graphical Aid to the Interpretation and Validation of Cluster Analysis"*, publicado en 1987 en el *Journal of Computational and Applied Mathematics*. En dicho trabajo, el autor propone una

representación gráfica denominada "silueta", diseñada para evaluar de forma visual e interpretativa la calidad de un agrupamiento. Matemáticamente, para cada punto i, la silueta se define como:

$$s(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}}$$

donde $a(i)$ representa la distancia promedio entre el punto i y los demás puntos dentro de su mismo clúster (medida de cohesión), mientras que $b(i)$ es la menor distancia promedio entre i y todos los puntos de cualquier otro clúster al que no pertenece (medida de separación). El valor de $s(i)$ se encuentra en el rango $[-1, 1]$: valores cercanos a 1 indican una asignación adecuada, cercanos a 0 indican ambigüedad (puntos en frontera), y valores negativos pueden sugerir mala asignación. El *ancho promedio de las siluetas* sirve como medida global de validez del agrupamiento, y se suele utilizar para comparar modelos o seleccionar el número óptimo de clústeres, eligiendo aquel que maximiza esta medida (Rousseeuw, 1987).

- **Índice de Calinski-Harabasz**

  Este indicador fue propuesto por Tadeusz Caliński y Jerzy Harabasz en su influyente trabajo titulado *"A dendrite method for cluster analysis"*, publicado en 1974 en la revista *Communications in Statistics*. El índice es una de las métricas más utilizadas para evaluar la calidad de particiones generadas mediante técnicas de agrupamiento no supervisado.

  Su fundamento se basa en una comparación directa entre la dispersión entre grupos y la dispersión dentro de los grupos. Por ende, el índice mide qué tan separadas están las agrupaciones entre sí y qué tan compactas son internamente. Esta relación se ajusta mediante factores de corrección basados en los grados de libertad, lo que permite hacer comparaciones justas incluso

cuando el número de clústeres varía. En términos matemáticos se tiene:

$$\text{CH} = \frac{\text{tr}(B_k)}{\text{tr}(W_k)} \cdot \frac{n-k}{k-1}$$

donde n es el número total de observaciones, k representa el número de clústeres, $\text{tr}(B_k)$ corresponde a la traza de la matriz de dispersión entre clústeres, y $\text{tr}(W_k)$ a la traza de la matriz de dispersión dentro de los clústeres. Esta formulación garantiza que se premie simultáneamente una alta separación entre clústeres y una baja variabilidad interna. Desde un punto de vista interpretativo, valores altos del índice Calinski-Harabasz indican una estructura de clústeres más definida, es decir, con fronteras claras y una asignación más consistente de los puntos dentro de cada grupo. (Caliński and Harabasz, 1974)

- **Índice de Davies-Bouldin**

  Fue desarrollado en 1979 por David L. Davies y Donald W. Bouldin en su artículo *"A Cluster Separation Measure"*, publicado en la revista *IEEE Transactions on Pattern Analysis and Machine Intelligence*. Este índice constituye una medida de validación interna para evaluar la calidad de particiones en análisis de clústeres, con base exclusivamente en la información intrínseca de los datos y sin necesidad de etiquetas externas.

  La lógica detrás del índice consiste en calcular, para cada clúster, su grado de similitud con el clúster más cercano, utilizando una relación entre la dispersión interna de los clústeres y la separación entre sus centroides. Formalmente, el índice se define como:

$$\text{DB} = \frac{1}{k} \sum_{i=1}^{k} \max_{j \neq i} \left(\frac{s_i + s_j}{d_{ij}} \right)$$

donde k es el número total de clústeres, s_i representa la dispersión interna del clúster i, el cual se calculada como la distancia euclidiana promedio de sus puntos al centroide y d_{ij} es la distancia euclidiana entre los centroides de los clústeres i y j. El término $\frac{s_i+s_j}{d_{ij}}$ refleja una medida de "similitud" entre dos clústeres, considerando cuán compactos son individualmente y cuán separados están entre sí. Luego, para cada clúster i, se toma el valor máximo de esta razón respecto a todos los demás clústeres, por ende, se identifica el peor caso de superposición relativa, y se promedia entre todos los clústeres.

Desde el punto de vista interpretativo, valores más bajos del índice Davies-Bouldin indican una mejor calidad de agrupamiento, ya que reflejan configuraciones con clústeres compactos y bien separados. El valor mínimo teórico es cero, lo que correspondería a clústeres completamente disjuntos y sin dispersión interna. Por lo tanto, se considera que la mejor partición del conjunto de datos es aquella que minimiza el valor del índice. (Davies and Bouldin, 1979)

3.2. Resultados

Al comparar los métodos de clasificación no supervisada con las métricas anteriores, se obtiene la siguiente tabla, donde **cal_har** hace referencia al Índice de Calinski-Harabasz y **dav_bou** al Índice de Davies-Bouldin.

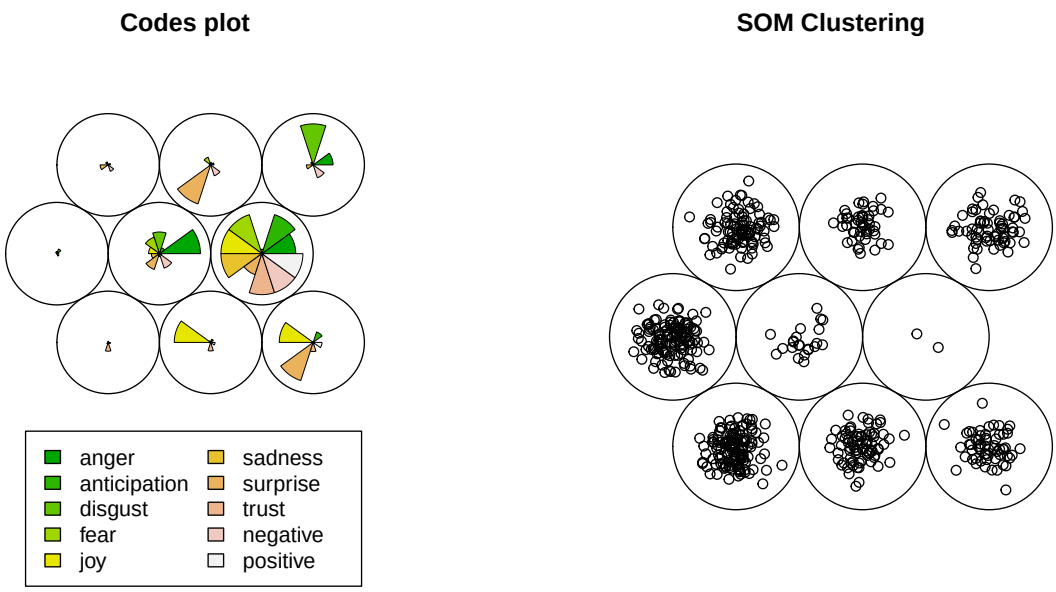
Tabla 6: Comparación de los métodos de clusterización.

método	**clusters**	**silhouette**	**cal_har**	**dav_bou**
kmeans	5	0.16273365	85.97041	1.986844
kmedians	5	0.09894534	81.14052	1.986909
jerarquico	5	0.19617423	171.65244	1.167912
dbscan	5	0.11451206	49.25579	1.663793
gmm	7	0.15317934	68.59909	2.482873
som	9	0.23831698	123.10330	1.470868
spectral	5	0.22614476	108.12655	1.601197
teigen	3	0.26243242	93.88190	1.319699
meanshift	5	0.31322985	76.44206	1.746621

Fuente: Elaboración propia.

Al observar los datos en la Tabla 6 y realizando una eliminación de métodos donde en cada iteración se elimina a los 3 peores en cada índice, se obtiene como resultado final la clusterización por medio de SOM. Se procede a tomar esta clusterización y observar sus resultados, primero graficando como se puede ver en la Figura 11

Figura 11: Visualización breve de los clusters.

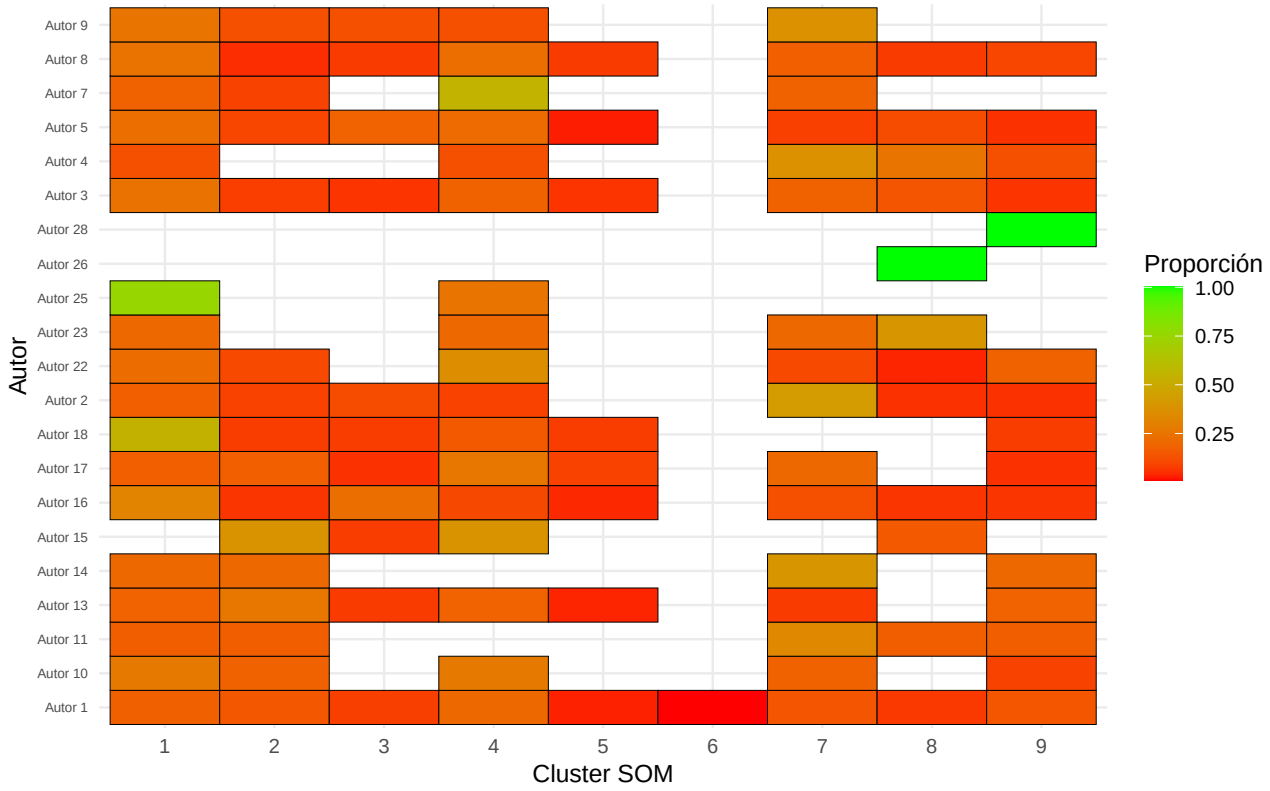


Fuente: Elaboración propia.

En esta figura hay mucha información resumida. Primero, al observar el lado de la derecha, se pueden notar visualmente solo la cantidad de observaciones en cada clúster, puesto que su posición no es determinada en este gráfico ya que viven en un espacio de 9 dimensiones. Lo único que realiza este gráfico es añadir puntos con ruido para poder observar la densidad de puntos, que por ejemplo, la primer columna de clústers son la que más observaciones tiene y por tanto, del lado izquierdo no se nota una presencia similar en las emociones de las observaciones.

Por otro lado, se nota que el clúster de la tercer columna y segunda fila solo tiene 2 observaciones, dando como resultado una similitud alta entre las emociones de las observaciones, ya que hay un margen más pequeño de diferencias. Ahora bien, en los demás clústers se nota una cantidad variada de observaciones, y hay una predominancia de alguna emoción en cada uno de estos clústers, lo que significaría que las observaciones en estos clústers tienen una predominancia en ese sentimiento.

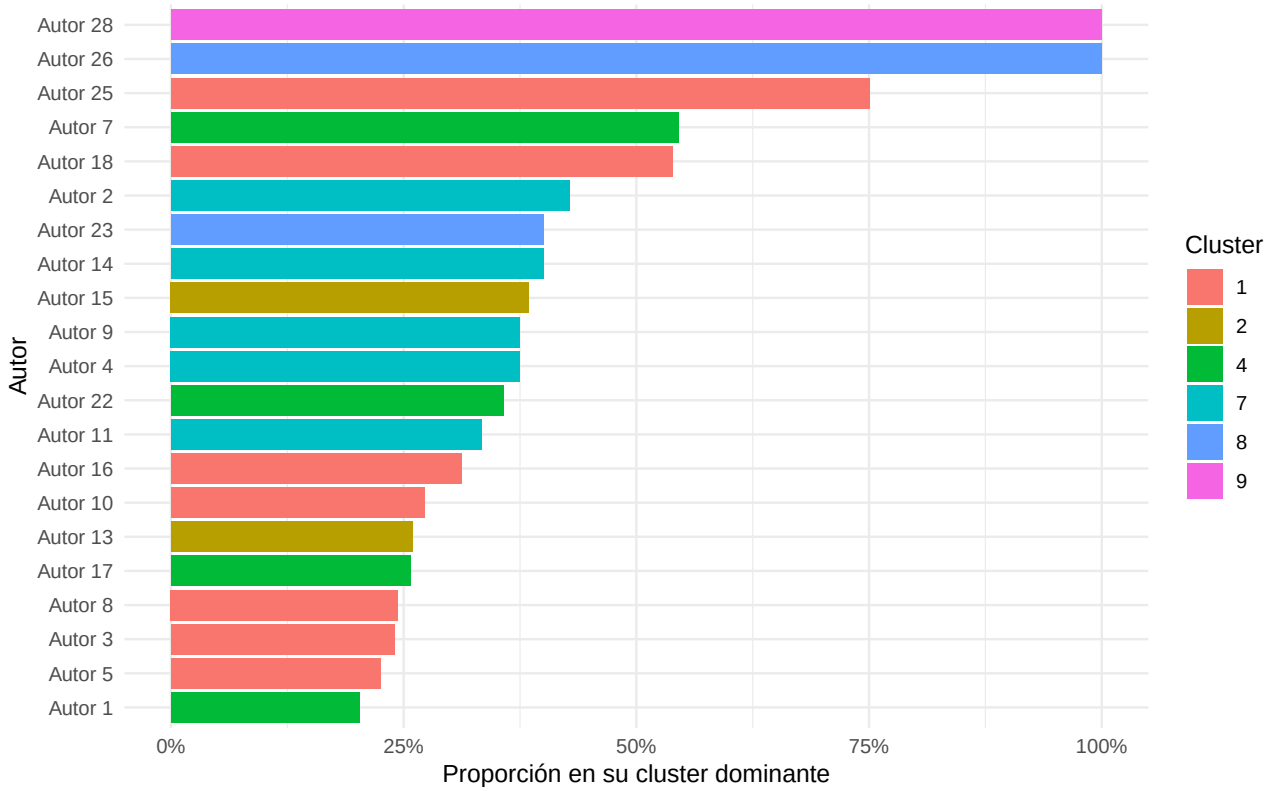
Figura 12: Proporción de mensajes de cada autor por clúster.



Fuente: Elaboración propia.

En el Figura 12 se pueden observar las proporciones de los mensajes de cada autor. Se evidencia que hay algunos autores que toda su presencia está en un cluster, ya que han mandado muy pocos mensajes. Aunque se puede observar presencia y no presencia de los autores, es mejor comparar con el clúster de mayor proporción, como se observa en la Figura 13.

Figura 13: Clúster dominante de cada autor.



Fuente: Elaboración propia.

En este gráfico, se nota claramente el clúster dominante de cada autor, donde a su vez se aprecia que en los clústers 3, 5 y 6 no hay algún autor que tenga la mayor cantidad de mensajes en él. Se pueden ver los dos autores mencionados anteriormente, con toda su presencia en un clúster, y se nota de la Figura 5 que los autores con más mensajes tienen una presencia baja en su clúster predominante, por la variedad de mensajes.

3.3. Conclusiones

Este proyecto permitió demostrar que los algoritmos de clasificación no supervisada pueden aplicarse con éxito al análisis de sentimientos en mensajes de texto, específicamente en conversaciones de WhatsApp de un grupo de estudiantes. A través del preprocesamiento textual, la vectorización de los mensajes y la implementación de distintos métodos de clusterización, fue posible identificar agrupamientos que reflejan patrones de contenido emocional en la comunicación escrita del grupo analizado.

La comparación entre modelos, basada en métricas internas como el coeficiente de silueta, el índice de Calinski-Harabasz y el índice de Davies-Bouldin, permitió evaluar objetivamente el rendimiento de cada algoritmo. Mediante una eliminación progresiva de los métodos con menor desempeño, se seleccionó el modelo de Mapas Auto-Organizativos como el más adecuado. Este algoritmo no solo obtuvo resultados competitivos en todas las métricas, sino que además ofreció una visualización estructurada que facilitó la interpretación de los clústeres.

El análisis de los clústeres generados por el SOM mostró que cada grupo tiende a presentar una emoción predominante, lo cual sugiere que el modelo logró capturar consistencias afectivas en los mensajes. A nivel individual, se observó que algunos autores concentran su participación en un único clúster, mientras que otros, especialmente quienes envían más mensajes, presentan una distribución más dispersa. Esto podría reflejar diferencias en estilo comunicativo o en la variedad emocional de sus intervenciones.

3.4. Limitaciones

Aunque el conjunto de datos fue exportado directamente desde WhatsApp y no presentaba valores faltantes, surgieron varias limitaciones durante el procesamiento y análisis. Una de las principales fue el formato del archivo exportado, que puede variar entre dispositivos o configuraciones, haciendo necesario un proceso manual de limpieza y estandarización que, si bien fue efectivo, no está libre de

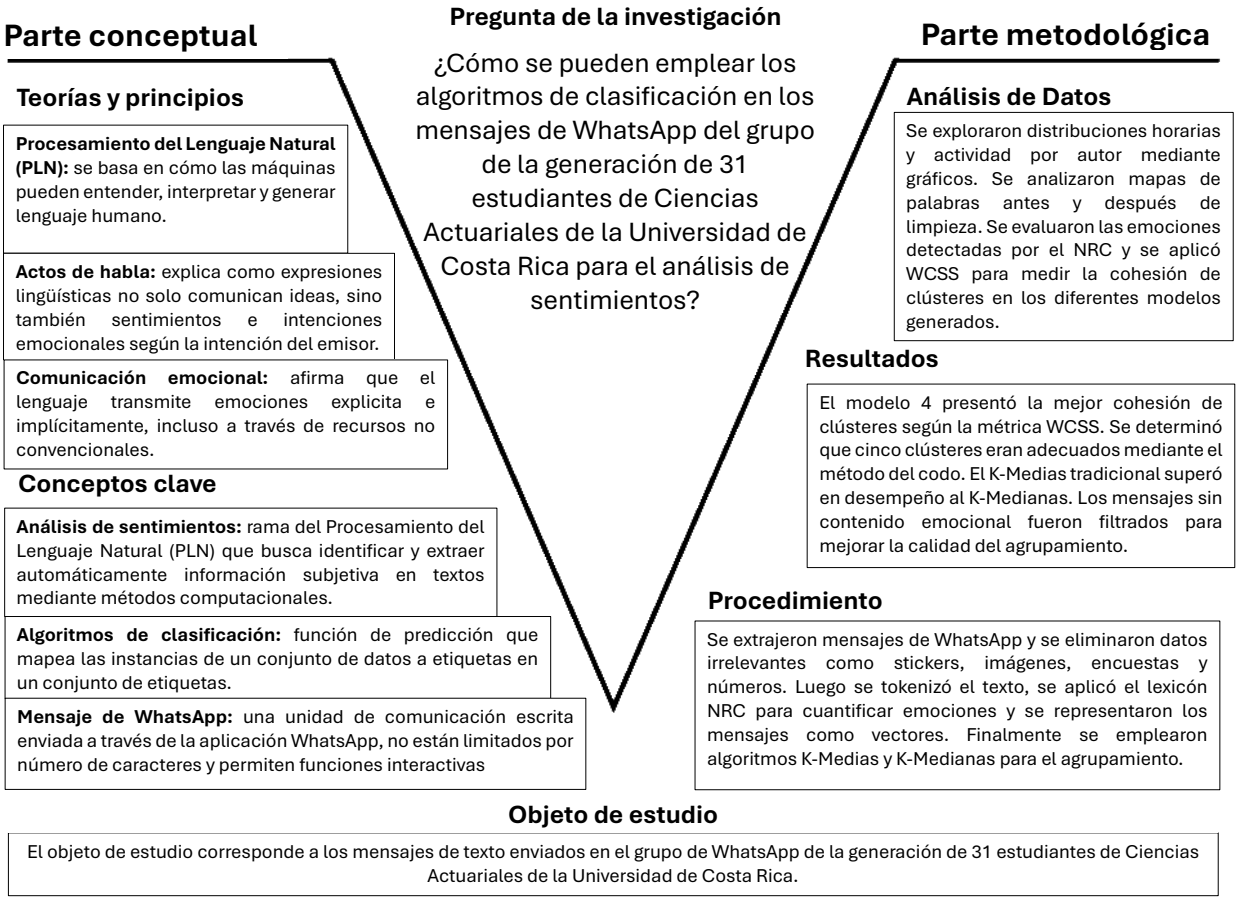
posibles pérdidas de estructura o contexto.

Otra limitación importante estuvo relacionada con el análisis de emociones. La mayoría de los diccionarios disponibles para este fin están en inglés, por lo que fue necesario traducirlos al español antes de aplicarlos. Este paso, aunque necesario, introduce posibles ambigüedades lingüísticas o pérdidas de significado emocional, ya que no todas las palabras tienen una correspondencia directa o equivalente en ambos idiomas.

Además, el análisis se realizó palabra por palabra, lo que implica una simplificación significativa frente al análisis del mensaje completo. Este enfoque permitió reducir la complejidad computacional y facilitar la implementación, pero limita la capacidad de capturar matices más sutiles, como la ironía o los cambios de tono dentro de un mismo mensaje. Trabajar con unidades textuales más amplias, como oraciones o párrafos completos, requeriría recursos más avanzados tanto a nivel computacional como de programación.

3.5. Uve de Gowin actualizada

Figura 14: Diagrama Uve de Gowin actualizada.



Fuente: Elaboración propia.

4. Referencias

Aggarwal, C. C. (2015). *Data Mining: The Textbook*. Springer International Publishing, Cham, Heidelberg, New York, Dordrecht, London.

Anónimo (s.f.). Métodos jerárquicos de análisis cluster. `https://www.ugr.es/~gallardo/pdf/cluster-3.pdf`. Universidad de Granada, capítulo 3.

Austin, J. L. (1962). *How to Do Things with Words*. Oxford University Press, Oxford.

Awan, A. A. (2024). ¿qué es la tokenización? `https://www.datacamp.com/blog/what-is-tokenization`.

Bishop, C. M. (2006). *Pattern Recognition and Machine Learning*. Springer, New York.

Caliński, T. and Harabasz, J. (1974). A dendrite method for cluster analysis. *Communications in Statistics*, 3(1):1–27.

Church, K. and de Oliveira, R. (2013). What's up with whatsapp? comparing mobile instant messaging behaviors with traditional sms. In *Proceedings of the 15th international conference on Human-computer interaction with mobile devices and services*, pages 352–361. ACM.

Comaniciu, D. and Meer, P. (2002). Mean shift: A robust approach toward feature space analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(5):603–619.

Cottrell, M., Olteanu, M., Rossi, F., and Villa-Vialaneix, N. (2018). Self-organizing maps, theory and applications. Technical report, Université Paris I, SAMM.

Davies, D. L. and Bouldin, D. W. (1979). A cluster separation measure. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-1(2):224–227.

Ester, M., Kriegel, H.-P., Sander, J., and Xu, X. (1996). A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining (KDD)*.

Hahsler, M., Piekenbrock, M., and Doran, D. (2019). dbscan: Fast density-based clustering with r. *Journal of Statistical Software*.

IBM (2022). Palabras vacías. `https://www.ibm.com/docs/es/db2/11.1.0?topic=configuration-stop-words`.

Jurafsky, D. and Martin, J. H. (1999). *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*. Prentice Hall, Upper Saddle River, NJ.

McLachlan, G. J. and Peel, D. (2000). *Finite Mixture Models*. Wiley, New York.

Meta Platforms, Inc. (2025). Whatsapp. `https://www.whatsapp.com/`.

Mohammad, S. M. (2021). Sentiment analysis: Detecting valence, emotions, and other affectual states from text. In Meiselman, H. L., editor, *Emotion Measurement, 2nd Edition*, pages 213–238. Woodhead Publishing.

Mohammad, S. M. and Turney, P. D. (2013). Crowdsourcing a word–emotion association lexicon. *Computational Intelligence*, 29(3):436–465.

Murel, J. and Kavlakoglu, E. (2023). ¿qué son el stemming y la lematización? `https://www.ibm.com/es-es/think/topics/stemming-lemmatization`.

Murel, J. and Kavlakoglu, E. (2024). ¿qué son los modelos de clasificación? https://www.ibm.com/es-es/think/topics/classification-models#:~:text=Los%20algoritmos%20de%20clasificaciÃşn%20se,y%20el%20filtrado%20de%20spam.

Oller Aznar, J. I. (2023). ¿qué son y para qué puedo usar las regex? https://jotelulu.com/blog/que-son-y-para-que-puedo-usar-las-regex/.

Rehman, S. U., Aziz, K., Fong, S., and Sarasvady, S. (2014). Dbscan: Past, present and future. In *2014 International Conference on Applications of Digital Information and Web Technologies (ICADIWT)*. IEEE.

Rousseeuw, P. J. (1987). Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, 20:53–65.

Sande, J. C. S. (2018). Análisis de sentimientos en twitter. Trabajo final de máster, Universitat Oberta de Catalunya.

Sawant, O., Kahna, M., and McAleese, S. (2023). ¿cuáles son las técnicas más efectivas para eliminar palabras vacías de los datos de texto? https://es.linkedin.com/advice/0/what-most-effective-techniques-removing-stop-words-a3ruf?lang=es.

Tibshirani, R. (2013). Hierarchical clustering. https://www.stat.cmu.edu/~ryantibs/datamining/lectures/06-clus3-marked.pdf. Carnegie Mellon University.

Trejos, J., Castillo, W., and González, J. (2021). *Análisis multivariado de datos: Métodos y aplicaciones*. Editorial Universidad de Costa Rica, San José, Costa Rica, primera edición digital revisada edition.

von Luxburg, U. (2007). A tutorial on spectral clustering. *Statistics and Computing*, 17(4):395–416.

Wehrens, R. and Buydens, L. M. C. (2007). Self- and super-organizing maps in r: The kohonen package. *Journal of Statistical Software*.

Zhou, Q., Zhao, K., and Zhang, Y. (2020). Dbscan clustering algorithm based on density. In *2020 7th International Forum on Electrical Engineering and Automation (IFEEA)*. IEEE.