Pontificia Universidad
JAVERIANA
Cali

ClientMinds - Optimización de la Experiencia del Cliente utilizando modelos de PLN

Autores

Jonathan Potes Blandón Código: 1'143.827.554
Obed García Quiroz Código: 1'052.415.577

Proyecto Aplicado para optar al título de
Magister en Ciencia de Datos

Director

Diego Luis Linares Ospina

Codirectora

Gloria Ines Alvarez Vargas

Facultad de Ingeniería y Ciencias
Maestria Ciencia de Datos
Santiago de Cali, Julio 07 2025

Tabla de contenido

Introducción 8

1. Definición del Problema **9**
1.1. Planteamiento del Problema 9
1.2. Formulación del problema 10

2. Objetivos **11**
2.1. Objetivo General 11
2.2. Objetivos Específicos 11

3. Alcance y justificación **12**
3.1. Alcance 12
3.2. Justificación 14

4. Marco teórico y antecedentes **15**
4.1. Marco teórico 15
4.1.1. Satisfacción, retención y conocimiento de clientes 15
4.1.2. Gestión de Relaciones con el Cliente 18
4.1.3. Procesamiento de Lenguaje Natural (NLP) 20
4.1.4. Modelos de Análisis de Sentimientos (SAM) 21
4.1.5. Modelos de Lenguaje a Gran Escala (LLM) 23
4.1.6. Soluciones Agénticas para Atención al Cliente 29
4.2. Antecedentes 30
4.2.1. Integración y construcción de IA con datos de CRM 30
4.2.2. Entendimiento del cliente a través de los datos de un CRM . 31
4.2.3. Modelos LLM 32
4.2.4. Modelos de Análisis de Sentimientos 32
4.2.5. Creación de ChatBot que utilice IA 33

5. Dataset Utilizado **35**
5.1. Descripción del Dataset Utilizado 35
5.2. Delimitación del Dominio de Aplicación 36
5.3. Análisis Exploratorio 37
5.3.1. Distribución porcentual por categoría e intención 37
5.3.2. Distribución del Número de Caracteres 38
5.3.3. Distribución de Tokens 38

6. Comprensión Emocional del Cliente a través de Modelos de Análisis de Sentimientos **40**
6.1. Limpieza y Preprocesamiento 40
6.1.1. Descripción del Dataset Inicial 40
6.1.2. Limpieza de los Datos 41

6.1.3. Construcción del Dataset para el Modelado 41
6.1.4. Preprocesamiento de los Datos 46
6.2. Modelado y desarrollo 47
6.2.1. Selección 47
6.2.2. Entrenamiento 48
6.2.3. Evaluación 50
6.3. Resultados y análisis 51
6.3.1. Análisis comparativo 51
6.3.2. Modelo Seleccionado 53
6.3.3. Conclusiones 56

7. Adaptación de Modelos LLM para Interacción con el Cliente 58
7.1. Limpieza y Preprocesamiento 58
7.1.1. Descripción del Dataset Inicial 58
7.1.2. Limpieza de Datos 59
7.1.3. Construcción del Dataset para el Modelado 60
7.2. Modelado y Desarrollo 61
7.2.1. Selección 61
7.2.2. Entrenamiento 63
7.3. Evaluación 67
7.3.1. Criterios de evaluación 67
7.3.2. Métricas de evaluación 67
7.4. Resultados y análisis 68
7.4.1. Evaluación con metricas clásicas 68
7.4.2. Evaluación complementaria con métricas modernas 68
7.4.3. Monitoreo del ajuste fino mediante QLoRA 70
7.4.4. Evaluación Comparativa de la Generación de Respuestas con y sin Sentimiento 72
7.4.5. Conclusiones 73

8. Desarrollo de un Chatbot como Plataforma de Atención al Cliente 75
8.1. Implementación del Chatbot Inteligente 75
8.2. Arquitectura General del Sistema 75
8.2.1. Componentes Principales 75
8.2.2. Tecnologías y Herramientas Utilizadas 76
8.3. Integración de los Modelos Desarrollados 76
8.3.1. Modelo de Análisis de Sentimientos 76
8.3.2. Modelo LLM para Generación de Respuestas 76
8.4. Flujo de Datos e Interacción con el Usuario 77
8.4.1. Proceso de Entrada y Clasificación Emocional 77
8.4.2. Generación de Respuesta Personalizada 78
8.4.3. Comparación y Alucinaciones: 79
8.4.4. Ejemplo de Interacción 80
8.5. Despliegue del Sistema 82
8.5.1. Entorno de Producción 82
8.5.2. Pruebas de Funcionamiento 82
8.6. Resultados y Observaciones 82
8.6.1. Evaluación Funcional 82
8.6.2. Evaluación Cualitativa con Encuesta 82

9. Reproducibilidad de los Resultados **84**
9.1. Estructura del Repositorio 84
9.2. Acceso al Dataset 84
9.3. Entorno de Ejecución 85
9.4. Consideraciones Finales 85

10. Conclusiones y Trabajos Futuros **86**
10.1. Conclusiones 86
10.2. Trabajos Futuros 88

Bibliografía **90**

Índice de cuadros

5.1. Agrupación de intenciones según la categoría del dataset 36

6.1. Comparación cualitativa de modelos 48
6.2. Comparación de Métricas Globales entre Modelos 51
6.3. Comparación de Métricas por Clase entre Modelos 52
6.4. Comparación de grilla de hiperparámetros antes y después de la mejora 55
6.5. Comparación de desempeño: Modelo Inicial vs. Modelo Mejorado . 55
6.6. Comparación de configuración óptima entre modelos 55

7.1. Resumen comparativo de los modelos LLM evaluados 63
7.2. Parrilla de búsqueda de hiperparámetros evaluada 65
7.3. Resultados de la evaluación de los modelos utilizando las métricas BLEU, ROUGE-L y Perplejidad. 68
7.4. Evolución de la pérdida durante el entrenamiento con QLoRA . . . 70
7.5. Comparación de métricas entre modelos con y sin variable de sentimiento 72

8.1. Criterios evaluados en la encuesta cualitativa 83

Índice de figuras

4.1. Clasificación de clientes en función de su permanecia y grado de satisfacción y lealtad . . . 17
4.2. Arquitectura Transformer basada en el modelo original de Vaswani et al. (2017) . . . 25

5.1. Distribución de registros por categoría . . . 37
5.2. Distribución de registros por intención . . . 37
5.3. Distribución del número de caracteres en los registros . . . 38
5.4. Distribución de recuentos de tokens para instrucciones antes de la limpieza. . . . 38
5.5. Distribución de recuentos de tokens para salidas antes de la limpieza. 39
5.6. Distribución de recuentos de tokens para instrucciones y salidas combinadas antes de la limpieza. . . . 39

6.1. Distribución de la población original versus la muestra balanceada utilizada para entrenamiento. . . . 43
6.2. Distribución de la clasificación automática del modelo preentrenado versus la clasificación manual definitiva. . . . 45
6.3. Matriz de Confusión - RNN . . . 52
6.4. Matriz de Confusión - BiLSTM . . . 52
6.5. Matriz de Confusión - BiGRU . . . 52
6.6. Accuracy - RNN . . . 53
6.7. Accuracy - BiLSTM . . . 53
6.8. Accuracy - BiGRU . . . 53
6.9. Pérdida - RNN . . . 53
6.10. Pérdida - BiLSTM . . . 53
6.11. Pérdida - BiGRU . . . 53

7.1. Diagrama del proceso de construcción del dataset . . . 60
7.2. Evaluación del modelo mediante métricas RAGAS (muestra de 500) 69
7.3. Evolución de la pérdida durante el entrenamiento con QLoRA . . . 71

8.1. Arquitectura general del sistema de atención al cliente basada en LLMs . . . 77
8.2. Procesamiento de informacion . . . 78
8.3. Interfaz del asistente ClientMinds con respuesta generada . . . 81

8.4. Evaluación cualitativa de respuestas generadas con y sin ingeniería de prompts 83

Listado de anexos

1. Carta de aceptación del director ——— 1

2. Carta de presentación de la propuesta ——— 1

Introducción

En el mundo empresarial actual, la gestión eficaz de las relaciones con los clientes (CRM, por sus siglas en inglés) se convierte en un componente esencial para el éxito y sostenibilidad de las organizaciones [7]. Sin embargo, muchas empresas enfrentan dificultades para comprender y mejorar la satisfacción del cliente, a pesar de tener acceso a abundantes datos en sus sistemas de CRM [2]. La principal limitación radica en la capacidad de extraer información significativa y aplicable de estos datos, lo que impide a las empresas identificar áreas de mejora y aplicar estrategias efectivas para retener clientes y proporcionar una experiencia satisfactoria [4].

Dada la problemática anterior y la oportunidad que tienen las organizaciones con los datos almacenados, este proyecto tuvo como objetivo desarrollar un sistema de Procesamiento del Lenguaje Natural (PNL) basado en un modelo de lenguaje a gran escala (LLM) para mejorar la comprensión y satisfacción del cliente. Este modelo no solo permite interpretar y responder a las solicitudes de los clientes de manera personalizada, sino que también analiza los sentimientos expresados en las interacciones y responden en consecuencia. Las estrategias utilizadas durante el desarrollo del sistema incluyen la limpieza y exploración de datos textuales, el ajuste fino de LLM preentrenados y la creación de un chatbot que integra estas capacidades. Este chatbot se creo de tal manera que no solo interactúa eficazmente con los clientes, sino que también deja registro de dichas interacciones con el propósito de generar insights valiosos, los cuales permiten hacer seguimiento a la satisfacción del cliente y tomar decisiones estratégicas basadas en su comportamiento.

Los resultados de este proyecto incluyen la creación de una base de datos depurada, obtenida a través de la limpieza y exploración de datos textuales provenientes de las interacciones con los clientes. La selección del modelo Llama2 como el mejor LLM para cumplir con el objetivo del proyecto y la creación de un chatbot que integra este modelo con una infraestructura tecnologica que interactúa con los clientes para resolver problemas y consultas, sirve como medio de comunicación y deja rgistro de dichas interacciones.

En resumen, este proyecto no solo representa un avance técnico en el procesamiento y análisis de datos de CRM para las organizaciones, sino que también responde a desafíos reales de la industria, proporcionando soluciones innovadoras y efectivas para la gestión de las relaciones con los clientes en el siglo XXI. En este contexto, el sistema de PNL, se consolida como una herramienta clave para mejorar la experiencia del cliente y fortalecer las relaciones cliente-empresa, gracias a la capacidad de los LLM para identificar sentimientos, ofrecer respuestas precisas y generar información valiosa, lo que facilita la implementación de acciones efectivas orientadas a la mejora continua en la interacción con los clientes.

Capítulo 1

Definición del Problema

1.1. Planteamiento del Problema

Las empresas enfrentan dificultades para comprender y mejorar la satisfacción del cliente, a pesar de tener acceso a abundantes datos en sus sistemas de Gestión de Relaciones con el Cliente (CRM) [1], [2]. Este problema es persistente y relevante en el entorno empresarial actual, donde la competencia es feroz y las expectativas de los consumidores continúan en aumento. La principal limitación radica en la capacidad de extraer información significativa y aplicable de estos datos, lo que impide a las empresas identificar áreas de mejora y aplicar estrategias efectivas para retener clientes y proporcionar una experiencia satisfactoria [3].

A pesar de recolectar y almacenar grandes cantidades de datos, las empresas carecen de herramientas analíticas avanzadas que les permitan comprender patrones y tendencias ocultas en la información [4]. Esta falta de herramientas puede llevar a que las organizaciones no logren aprovechar todo el potencial de la información recopilada. Como resultado, muchas empresas se encuentran atrapadas en un ciclo donde los datos se convierten en un recurso subutilizado, en lugar de ser un activo estratégico que impulse decisiones informadas y oportunas. La incapacidad para extraer insights valiosos limita la habilidad de las empresas para anticipar las necesidades y preferencias de sus clientes.

La falta de una comprensión profunda de los datos y la incapacidad para anticipar y responder de manera proactiva a sus demandas son barreras persistentes que las empresas deben abordar urgentemente [5]. En este sentido, la interpretación errónea de la información o la falta de análisis adecuado puede llevar a decisiones mal fundamentadas que, en última instancia, afectan la lealtad del cliente y su satisfacción general. Las empresas que no logran comprender las emociones, deseos y comportamientos de sus clientes se enfrentan al riesgo de perder competitividad en un mercado que se encuentra en constante evolución.

Mejorar la satisfacción y lealtad del cliente es crucial en un entorno empresarial donde la retención de clientes y la mejora continua de la experiencia del cliente son

esenciales para el éxito a largo plazo. La relación entre la satisfacción del cliente y el rendimiento organizacional se ha documentado ampliamente, evidenciando que las empresas que logran cultivar relaciones sólidas con sus clientes experimentan un crecimiento sostenido y una rentabilidad superior [6]. Por lo tanto, las organizaciones deben adoptar enfoques proactivos que les permitan no solo recolectar datos, sino también analizarlos y transformarlos en estrategias efectivas que promuevan una mejor experiencia del cliente.

Además, el uso de técnicas avanzadas, como el procesamiento de lenguaje natural y análisis de sentimientos, se presenta como una oportunidad clave para que las empresas obtengan una ventaja competitiva. Estos métodos permiten una comprensión más rica de las interacciones del cliente, permitiendo a las empresas no solo reaccionar a sus necesidades, sino también anticiparse a ellas. Sin embargo, el desafío persiste: ¿cómo pueden las empresas superar las limitaciones actuales en sus capacidades de análisis de datos y adoptar soluciones que realmente mejoren la satisfacción del cliente?

En resumen, las empresas están en un punto crítico donde deben reevaluar sus estrategias de CRM para integrar tecnologías avanzadas que faciliten una comprensión más profunda de sus clientes. A medida que el panorama empresarial se vuelve más complejo, aquellas organizaciones que logren adaptarse y aplicar técnicas innovadoras para la gestión de las relaciones con los clientes estarán mejor posicionadas para enfrentar los desafíos del futuro y asegurar su sostenibilidad en el mercado.

1.2. Formulación del problema

De acuerdo al planteamiento del problema surge la necesidad de responder los siguientes interrogantes:

¿Cómo puede el desarrollo e implementación de un enfoque integral basado en herramientas analíticas avanzadas mejorar la comprensión y la satisfacción del cliente en las empresas?. ¿Cómo pueden las técnicas de afinamiento preciso de modelos LLM preentrenados mejorar la capacidad de las empresas para analizar los datos del cliente y entender sus necesidades y preferencias?. ¿Cómo se puede procesar y analizar eficientemente el texto de las interacciones de los clientes para mejorar la satisfacción del cliente?. ¿De qué manera puede un chatbot, utilizando los modelos entrenados, mejorar la interacción personalizada con los clientes y resolver adecuadamente sus problemas y consultas?. ¿Cuáles son las métricas adecuadas para evaluar la efectividad de los modelos entrenados y del chatbot en términos de satisfacción y lealtad del cliente?.

Capítulo 2

Objetivos

2.1. Objetivo General

Desarrollar un ChatBot para la atención de consultas de usuarios aplicando modelos de lenguaje de gran tamaño (LLM) y análisis de sentimientos.

2.2. Objetivos Específicos

- Aplicar técnicas de exploración y limpieza de datos textuales para utilizarlos como base en modelos de procesamiento de lenguaje natural que mejoren la experiencia del cliente y descubran patrones que describan su comportamiento.
- Desarrollar un modelo de análisis de sentimientos para identificar el estado emocional del cliente en cada interacción con el sistema, proporcionando insights que ayuden a comprender su comportamiento ante diversas situaciones y faciliten acciones específicas para satisfacer sus necesidades de manera efectiva.
- Incorporar técnicas de fine-tuning en modelos LLM preentrenados para capacitar al sistema en el análisis inteligente y automatizado de las interacciones del cliente, permitiendo que el modelo LLM no solo identifique tendencias, preferencias y áreas de fricción que afecten su satisfacción y lealtad hacia la marca, sino también que pueda responder de manera adecuada a cada interacción con el cliente.
- Crear un chatbot que integre los modelos desarrollados para interactuar de forma personalizada con los clientes, resolver efectivamente sus problemas y consultas, y servir como base del sistema de procesamiento de lenguaje natural y principal medio de comunicación con el cliente.

Capítulo 3

Alcance y justificación

3.1. Alcance

A continuación se presenta el alcance para cada objetivo específico, definiendo qué elementos son parte del compromiso adquirido al realizar el proyecto y qué elementos quedan por fuera del proyecto y no hay compromiso de entregarlos.

1. Objetivo 1: Aplicar técnicas de exploración y limpieza de datos textuales para utilizarlos como base en modelos de procesamiento de lenguaje natural que mejoren la experiencia del cliente y descubran patrones que describan su comportamiento.

1.1. Dentro del alcance:

- Análisis y limpieza de los datos.
- Reporte de los resultados obtenidos.
- Construcción de datasets con los datos depurados.

1.2. Fuera del alcance:

- Desarrollo de herramientas de análisis y limpieza de los datos no mencionadas en el proyecto.
- Integración con sistemas externos no especificados previamente.

2. Objetivo 2: Desarrollar un modelo de análisis de sentimientos para identificar el estado emocional del cliente en cada interacción con el sistema, proporcionando insights que ayuden a comprender su comportamiento ante diversas situaciones y faciliten acciones específicas para satisfacer sus necesidades de manera efectiva.

2.1. Dentro del alcance:

- Ajuste de modelos de análisis de sentimientos.
- Entrenamiento de los modelos con el dataset creado para tal fin.
- Evaluación y selección de los modelos entrenados.
- Implementación en el ChatBot del modelo seleccionado.

2.2. Fuera del alcance:

- Desarrollo de nuevas herramientas de análisis de texto no mencionadas en el proyecto.
- Integración con sistemas externos no especificados previamente.

3. Objetivo 3: Incorporar técnicas de fine-tuning en modelos LLM preentrenados para capacitar al sistema en el análisis inteligente y automatizado de las interacciones del cliente, permitiendo que el modelo LLM no solo identifique tendencias, preferencias y áreas de fricción que afecten su satisfacción y lealtad hacia la marca, sino también que pueda responder de manera adecuada a cada interacción con el cliente.

3.1. Dentro del alcance:

- Ajuste de modelos LLM preentrenados.
- Entrenamiento de los modelos con el dataset creado para tal fin.
- Evaluación y selección de los modelos entrenados.
- Implementación en el ChatBot del modelo seleccionado.

3.2. Fuera del alcance:

- Desarrollo de nuevas herramientas de análisis de texto no mencionadas en el proyecto.
- Diseño de nuevos LLMs.
- Integración con sistemas externos no especificados previamente.

4. Objetivo 4: Crear un chatbot que integre los modelos desarrollados para interactuar de forma personalizada con los clientes, resolver efectivamente sus problemas y consultas, y servir como base del sistema de procesamiento de lenguaje natural y principal medio de comunicación con el cliente.

4.1. Dentro del alcance:

- Desarrollo e integración de un chatbot utilizando los modelos de LLM y análsis de sentimientos seleccionados.
- Pruebas y ajustes del chatbot para asegurar su funcionalidad y eficacia.

4.2. Fuera del alcance:

- Desarrollo de interfaces de usuario adicionales.
- Mantenimiento y soporte técnico del chatbot post-implementación.

3.2. Justificación

En el contexto actual, donde las empresas gestionan grandes volúmenes de datos a través de sistemas de Gestión de Relaciones con el Cliente (CRM), uno de los principales desafíos no radica en la falta de información, sino en la dificultad para convertir esos datos en conocimiento útil y accionable. La ausencia de herramientas analíticas avanzadas limita la capacidad de identificar patrones, emociones y comportamientos en las interacciones con los clientes, lo que obstaculiza la toma de decisiones estratégicas orientadas a mejorar su experiencia, satisfacción y fidelización.

Frente a esta necesidad, el presente proyecto propone el desarrollo de un sistema conversacional inteligente, apoyado en técnicas de procesamiento de lenguaje natural, modelos de análisis de sentimientos y modelos de lenguaje de gran tamaño (LLM), con el fin de transformar las interacciones textuales en una fuente rica de información. Esta solución no solo responde a consultas de manera automatizada y personalizada, sino que también permite comprender el estado emocional del cliente, identificar tendencias en sus comportamientos y adaptar las respuestas según el contexto, todo ello con base en datos reales.

Además, el proyecto se sustenta en herramientas tecnológicas de acceso abierto, lo cual demuestra su viabilidad y escalabilidad para organizaciones que deseen implementar soluciones basadas en inteligencia artificial sin incurrir en altos costos. Al aplicar metodologías propias de los proyectos de ciencia de datos —como la exploración, limpieza, modelado y evaluación—, se garantiza un enfoque riguroso para la generación de valor a partir de los datos.

En suma, este proyecto responde a una necesidad concreta del entorno empresarial moderno: mejorar la atención al cliente a través de la automatización inteligente y el análisis emocional, contribuyendo al diseño de experiencias más cercanas, eficientes y centradas en el usuario. Además, establece una base sólida para futuras mejoras analíticas y evoluciones tecnológicas en el ámbito del servicio al cliente.

Capítulo 4

Marco teórico y antecedentes

Con el propósito de establecer los fundamentos teóricos del proyecto, a continuación, se exponen los conceptos básicos en los que se sustenta. Inicialmente, se aborda la definición de satisfacción y retención del cliente, así como su interrelación con el conocimiento del cliente. Luego, se introduce el concepto de Gestión de Relaciones con el Cliente (CRM, por sus siglas en inglés), donde se define el CRM y se detallan sus componentes, destacando su vínculo con la satisfacción y retención de los clientes. Además, se exploran los desafíos que enfrentan las organizaciones al implementar estos sistemas, lo que a menudo conduce a la falta de cumplimiento del objetivo con el que es implementado un CRM.

Posteriormente se presenta la definición del Procesamiento de Lenguaje Natural (NLP) y de dos de sus técnicas avanzadas, a saber, los Modelos de Lenguaje a Gran Escala (LLM) y el Análisis de Sentimientos (AS), las cuales serán utilizadas durante la realización de este proyecto, con el objetivo de proveerle a las organizaciones herramientas para superar el no cumplimiento del objetivo mencionado previamente. Finalmente se presenta una descripción general de artículos encontrados en la literatura, que están directamente relacionados con el proyecto en cuestión y se describe dicha relación, esto con el objetivo de poder identificar los antecedentes que abordan el mismo problema o un problema similar y así tener una base teórica sólida durante el abordaje del proyecto.

4.1. Marco teórico

4.1.1. Satisfacción, retención y conocimiento de clientes

En el entorno empresarial actual, la satisfacción y retención de los clientes se han convertido en pilares fundamentales para el cumplimiento de los objetivos estratégicos de las organizaciones, particularmente en lo que respecta a la rentabilidad. Según Guadarrama Tavira y Rosales Estrada “una gestión efectiva de las relaciones con los clientes, centrada en satisfacer sus necesidades y expectativas, no solo fortalece el vínculo con el cliente, sino que también es crucial para alcanzar la rentabilidad deseada por la organización” [7].

A continuación, se definen los conceptos de satisfacción y retención de clientes, y se menciona cómo influyen en la rentabilidad empresarial, adicionalmente se muestra como estos dos conceptos se relacionan con el conocimiento del cliente y la gestión de relaciones del mismo.

3.1.1.1. Satisfacción de clientes

La satisfacción del cliente se define como el grado en que las expectativas del cliente son alcanzadas o superadas por un producto o servicio, lo que resulta en una experiencia positiva y una mayor probabilidad de lealtad y repetición de compra. Guadarrama Tavira y Rosales Estrada enfatizan que "la satisfacción emerge como un valor decisivo para el logro de la rentabilidad deseada por la organizaciónz que una gestión adecuada de las relaciones con los clientes ayuda a cubrir sus expectativas de calidad, lo cual está directamente relacionado con una mayor satisfacción y conlleva a generar lealtad con el cliente, posibilitando así su retención [7].

3.1.1.2. Retención de clientes

La retención del cliente se define como la capacidad de una empresa para mantener a sus clientes a lo largo del tiempo mediante la satisfacción continua de sus necesidades y expectativas. Según Guadarrama Tavira y Rosales Estrada, "la retención de clientes es una medida esencial para la sostenibilidad y rentabilidad de las organizaciones, ya que mantener a un cliente existente es más rentable que adquirir uno nuevo"[7].

3.1.1.3. Conocimiento de clientes

Para satisfacer y retener al cliente es de vital importancia conocerlo, y para esto, es fundamental establecer un diálogo continuo con él. Como mencionan Guadarrama Tavira y Rosales Estrada, “este intercambio constante de información permite a las empresas comprender mejor las necesidades y preferencias de sus clientes” [7]. De tal manera que al entender qué es lo que el cliente valora, las organizaciones podrán empezar a conocer realmente a sus clientes y llegar hasta el punto de clasificarlos de acuerdo con diferentes criterios y planteamientos como los que se presentan a continuación:

Los clientes de nivel "platino"	Son los clientes más rentables de la empresa con una alta tasa de compra y poco sensibles al precio. Hay que averiguar qué necesidades tienen para darles nuevos ofrecimientos y mantener su compromiso con la empresa.
Los clientes de nivel "oro"	Dan una alta rentabilidad, aunque inferior a los de nivel "platino". Desean continuos descuentos sobre el precio y no son tan leales como aquellos, pues suelen minimizar el riesgo comparando a varios proveedores.
Los clientes de nivel "hierro"	Son clientes que dan "volumen" (cuota de mercado) a la empresa, pero provocan mayores gastos, menor rentabilidad y no son totalmente leales.
Los clientes de nivel "plomo"	Son aquellos clientes que cuestan dinero a la empresa y no son leales.

(a) Clasificación piramidal de los clientes según Guadarrama Tavira y Rosales Estrada [7]

Etiqueta A Los recién incorporados.	Ofrecen información sobre los atributos del valor más apreciado, debido a que al iniciarse una relación de compra se pone mayor atención en los atributos específicos del producto o servicio ofrecido.
Etiqueta B o C	Llevan más tiempo de relación con la empresa y pueden aportar información de cómo reforzar las estrategias dirigidas a estrechar los lazos de relación con el cliente.
Etiqueta D Clientes que se han marchado	Aportan todo tipo de información, sobre todo se debe aprender de ellos como evitar su deceso. Conociendo y corrigiendo estas deficiencias, se evitará que se vayan los clientes, y se reforzará su relación con la empresa.

(b) Clasificación de clientes en función de su permanencia con la empresa según Guadarrama Tavira y Rosales Estrada [7]

Clasificación	Comportamiento
Cliente prescriptor	Está satisfecho y mantiene unas relaciones cordiales.
Cliente oportunista	Satisfecho, pero piensa que puede encontrar algo mejor.
Cautivo	Cliente descontento, se encuentra atrapado por nuestras condiciones y le resulta caro cambiar de proveedor. Es vengativo y destructor.
Destructor	Busca alternativas pensando que cualquiera puede ser mejor, está descontento y genera publicidad negativa.

(c) Clasificación de clientes en función del grado de satisfacción y lealtad según Guadarrama Tavira y Rosales Estrada [7]

Figura 4.1: Clasificación de clientes en función de su permanecia y grado de satisfacción y lealtad

Con una clasificación como la mencionada anteriormente, la organización estará mejor equipada para abordar las necesidades de sus clientes de manera personalizada, lo que se traducirá en una mayor satisfacción y retención. Sin embargo, para llevar a cabo esta clasificación de manera efectiva, es imprescindible establecer un intercambio constante de información con el cliente y registrar estos datos. Es en este punto donde surge la necesidad de implementar una adecuada Gestión de Relaciones con el Cliente (CRM por sus siglas en inglés: Customer Relationship Management) y contar con un sistema que respalde esta gestión. A continuación,

profundizaremos un poco más sobre esta estrategia.

4.1.2. Gestión de Relaciones con el Cliente

3.1.2.1. ¿Qué es un CRM?

CRM por sus siglas en inglés (Customer Relationship Management), es una estrategia de negocios que integra procesos y funciones internas y redes externas, apoyándose en tecnología y métodos científicos, para identificar nuevos clientes, construir relaciones duraderas con los clientes, retener a los clientes satisfaciendo sus necesidades y, finalmente, reducir su tasa de abandono, minimizando al mismo tiempo los costos de marketing y servicio al cliente [8].

3.1.2.2. ¿Cuál es el objetivo de un CRM?

Según Soumaya Lamrhari, Hamid El Ghazi y otros autores, el objetivo de un CRM "es el desarrollo y optimización de la relación con el cliente" [8]. Este enfoque subraya la importancia de establecer vínculos sólidos y duraderos con los clientes, lo que a su vez puede conducir a una mayor satisfacción del cliente y retención, así como a un incremento en la rentabilidad para la empresa.

3.1.2.3. Componentes de un CRM

De acuerdo con Soumaya Lamrhari, Hamid El Ghazi y otros investigadores, un CRM se compone de tres elementos que trabajan de manera conjunta para alcanzar el propósito mencionado anteriormente. A continuación, se detallan cada uno de estos componentes.

3.1.2.3.1. CRM Operacional

El CRM Operacional "se centra en la gestión diaria de la relación con el cliente. Su objetivo es abarcar todas las tareas que impliquen un contacto directo con los clientes. Proporciona soporte para diferentes procesos comerciales, incluidos servicios al cliente, gestión de pedidos, ventas y automatización de marketing. Todas las interacciones con el cliente se registran en el almacén de datos de la empresa con fines de análisis y seguimiento" [8].

3.1.2.3.2. CRM Colaborativo

El CRM colaborativo "es responsable de las interacciones con los clientes finales, como interacciones personales, sitios web, centros de llamadas, correo y ventas directos. Se ocupa de la sincronización e integración de los puntos de contacto de interacción con el cliente, así como de canales de comunicación como correo electrónico, teléfono, web, fax, para referenciar a los clientes de manera consistente y sistemática, así como permitir una comunicación intensiva y flexible entre clientes y empresas" [8].

3.1.2.3.3. CRM Analítico

El CRM analítico "proporciona un nivel más profundo de inteligencia. Su objetivo es analizar los datos de los clientes generados en gran medida por CRM operativo y colaborativo utilizando herramientas analíticas (como la Inteligencia Artificial) que abarcan técnicas avanzadas de estadística/aprendizaje automático, minería de textos y minería de datos web. Este análisis de datos se aprovecha para mejorar el servicio al cliente al descubrir información comercial oculta que de otro modo no se puede obtener con reglas comerciales estándar" [8].

3.1.2.4. Relación de un CRM con la satisfacción y retención de clientes

Como se mencionó anteriormente, un CRM es una herramienta fundamental para gestionar las interacciones y relaciones con los clientes de manera eficiente y personalizada. El CRM juega un papel crucial tanto en la satisfacción como en la retención de clientes por los siguientes motivos:

- Satisfacción del Cliente: Un sistema CRM permite a las empresas recopilar y analizar datos sobre las interacciones de los clientes, lo que facilita una comprensión profunda de sus necesidades y preferencias. Esta información permite ofrecer servicios más personalizados y adecuados, mejorando la experiencia del cliente. La implementación de un CRM contribuye significativamente a aumentar la satisfacción del cliente al proporcionar un servicio más eficiente y centrado en el cliente [8].

- Retención del Cliente: La gestión efectiva de las relaciones con los clientes a través de un CRM permite identificar y resolver problemas antes de que resulten en la pérdida de clientes. Al mejorar la satisfacción del cliente mediante un servicio más personalizado y eficiente, un CRM ayuda a aumentar la lealtad del cliente y, en consecuencia, su retención. Según el estudio, la satisfacción del cliente actúa como mediador en la relación entre la implementación de CRM y la retención del cliente. Esto significa que un manejo eficaz del CRM puede conducir indirectamente a una mayor retención de clientes al mejorar su satisfacción [8].

En resumen, la integración y el uso efectivo de un sistema CRM no solo mejoran la satisfacción del cliente, sino que también juegan un papel crucial en la retención de clientes, asegurando así el crecimiento y la sostenibilidad a largo plazo de la empresa.

3.1.2.5. Desafíos actuales en el análisis de datos de CRM en las organizaciones

Como se mencionó anteriormente, el objetivo principal detrás de la implementación de un CRM en una empresa es el desarrollo y la optimización de las relaciones con los clientes. Para lograr este objetivo, es esencial la implementación de los tres componentes del CRM. Sin embargo, la omisión del CRM analítico en muchas empresas, que tienden a enfocarse únicamente en el CRM operativo y colaborativo,

limita significativamente el alcance completo de esta meta y el potencial inherente de los datos almacenados en él. Esta carencia no es casual, sino que se debe a varios desafíos a los que se enfrentan las organizaciones en la actualidad. A continuación, se detallarán los principales desafíos identificados.

1. Prerrequisitos técnicos y acceso a datos de calidad: La efectividad de la IA en CRM depende de tener acceso a conjuntos de datos extensos y de alta calidad, así como de la infraestructura tecnológica adecuada para procesar esos datos de manera eficiente [9].

2. Integración con plataformas y bases de datos existentes: La integración de aplicaciones de IA en sistemas CRM requiere una integración perfecta con las plataformas y bases de datos existentes, lo que puede presentar desafíos técnicos y operativos, especialmente en entornos de datos complejos [9].

3. Definición de objetivos precisos: Es crucial definir objetivos precisos para los algoritmos de IA en el contexto de CRM, lo cual puede ser complicado debido a los objetivos implícitos y difíciles de cuantificar en el ámbito del marketing y las ventas [9].

4. Resistencia al cambio: La resistencia al cambio es común durante la implementación de la IA en CRM, ya que implica una colaboración estrecha entre agentes humanos y la IA, lo que requiere un equilibrio único en los roles y responsabilidades [9].

5. Costo de implementación de IA en el CRM: La integración de tecnologías de IA en los sistemas CRM puede requerir una inversión significativa en términos de adquisición de software, hardware especializado, capacitación de personal y mantenimiento continuo. Esta inversión financiera puede representar una barrera para algunas organizaciones, especialmente para aquellas con recursos limitados, lo que puede obstaculizar la adopción generalizada de la IA en CRM.

4.1.3. Procesamiento de Lenguaje Natural (NLP)

Una tecnología fundamental para abordar los desafíos mencionados previamente es el Procesamiento de Lenguaje Natural (NLP). Esta tecnología mejora la eficiencia y efectividad del CRM al comprender y procesar el lenguaje humano, aprovechando al máximo los datos almacenados, proporcionando valiosas herramientas e insights que amplían el conocimiento del cliente, lo que se traduce en una mayor satisfacción y retención de este. A continuación, se describe un poco más a fondo este tema.

3.1.3.1. ¿Qué es NLP?

El procesamiento del lenguaje natural NLP por sus siglas en inglés (Natural Language Processing) es un enfoque interdisciplinario del aprendizaje automático que combina la informática, la inteligencia artificial y la lingüística con el objetivo de enseñar el lenguaje natural a las máquinas. La idea no es solo dotar a los ordenadores de la capacidad de entender, procesar, simular y generar lenguaje humano, sino también permitir conversaciones naturales con los humanos. La traducción automática, los sistemas de preguntas y respuestas, el análisis de sentimientos, la clasificación de textos y muchas otras aplicaciones se pueden realizar a través de la tecnología NLP [10].

3.1.3.2. Técnicas de NLP relevantes para el proyecto

Si bien existen diversas técnicas de Procesamiento del Lenguaje Natural (PLN), a continuación se describen las dos técnicas que se emplearán en el proyecto de manera general y en los artículos científicos mostrados en los antecedentes, se detalla más a fondo cada una de estas técnicas.

4.1.4. Modelos de Análisis de Sentimientos (SAM)

El análisis de sentimientos (*Sentiment Analysis*) es una rama del procesamiento del lenguaje natural (PLN) que tiene como objetivo identificar, extraer y clasificar automáticamente las opiniones expresadas en un texto, especialmente con respecto a una entidad específica como un producto, servicio, evento o política pública. Esta técnica permite determinar si el sentimiento general del texto es positivo, negativo o neutro, y se ha convertido en una herramienta clave en contextos como el análisis de redes sociales, la atención al cliente, la investigación de mercado y la gestión de la reputación en línea [11].

Definición

Un modelo de análisis de sentimientos es una solución computacional entrenada para identificar patrones emocionales en textos escritos. Estos modelos son especialmente útiles para procesar grandes volúmenes de datos no estructurados generados por los usuarios en plataformas como Twitter, Facebook, foros en línea, blogs, y sistemas de reseñas. Su implementación permite a organizaciones privadas y públicas comprender mejor las actitudes del público, detectar tendencias, anticiparse a crisis de imagen o evaluar la aceptación de productos o políticas [12].

Aprendizaje supervisado

La mayoría de los modelos modernos de análisis de sentimientos están basados en técnicas de *aprendizaje supervisado*, un enfoque del aprendizaje automático en el que el modelo se entrena utilizando un conjunto de datos etiquetado, es decir, ejemplos de entrada con sus correspondientes salidas deseadas. En este contexto, los textos están etiquetados con su categoría de sentimiento (por ejemplo, positivo,

negativo o neutro), lo que permite que el algoritmo aprenda las características del lenguaje asociadas a cada clase [13].

Entre los algoritmos supervisados comúnmente utilizados para el análisis de sentimientos se encuentran los clasificadores bayesianos, máquinas de vectores de soporte (SVM), árboles de decisión y redes neuronales. Estos modelos pueden captar patrones semánticos, sintácticos y contextuales presentes en el texto, lo que los hace adecuados para la tarea de categorización emocional.

Redes Neuronales Recurrentes (RRN)

Las Redes Neuronales Recurrentes (RRN o RNN por sus siglas en inglés) representan una arquitectura de red especialmente diseñada para manejar datos secuenciales, como los textos naturales. A diferencia de las redes neuronales tradicionales que procesan los datos de forma independiente, las RRN introducen ciclos en su estructura, lo que les permite mantener una *memoria* interna de los estados anteriores. Esta capacidad de modelar dependencias temporales o contextuales es fundamental en tareas como la traducción automática, el reconocimiento de voz y, por supuesto, el análisis de sentimientos [14].

En el análisis de sentimientos, las RRN son capaces de interpretar cómo se desarrolla una opinión a lo largo de una oración, reconociendo el papel crucial que desempeñan la posición de las palabras, las negaciones y las expresiones compuestas. Por ejemplo, pueden distinguir entre "me gustó mucho" y "no me gustó", dos frases similares en forma pero opuestas en contenido emocional.

BiGRU: una variante eficiente para la clasificación de sentimientos

Dentro del marco de las redes neuronales recurrentes, se encuentran variantes como las redes *GRU* (Gated Recurrent Unit) y su versión bidireccional, *BiGRU*, que mejoran la capacidad de aprendizaje de dependencias a largo plazo y aceleran el proceso de entrenamiento al reducir la complejidad computacional. Las BiGRU procesan la información en ambas direcciones (hacia adelante y hacia atrás), lo cual permite capturar de manera más completa el contexto de cada palabra en la oración. Esta característica las convierte en modelos altamente efectivos para tareas de clasificación de sentimientos, donde el significado de una expresión puede depender tanto de las palabras anteriores como de las siguientes [15].

Importancia práctica y relevancia actual

Actualmente, el uso de modelos basados en redes neuronales para el análisis de sentimientos representa un enfoque de vanguardia, al ofrecer resultados más precisos que los métodos tradicionales. Gracias a su capacidad para aprender representaciones complejas del lenguaje, estos modelos permiten generar análisis más matizados y robustos, adaptándose incluso a dominios específicos o jergas propias de determinadas comunidades en línea.

En resumen, los modelos de análisis de sentimientos combinan técnicas de aprendizaje automático, procesamiento de lenguaje natural y estructuras neurona-

les avanzadas como las RRN y BiGRU para proporcionar una solución tecnológica potente frente al desafío de entender emocionalmente grandes volúmenes de texto. Su implementación estratégica facilita la toma de decisiones informadas en diversos ámbitos como el marketing, la política, la gestión pública y el servicio al cliente.

4.1.5. Modelos de Lenguaje a Gran Escala (LLM)

Los Modelos de Lenguaje de Gran Escala (LLM) son un tipo de inteligencia artificial que utiliza técnicas de aprendizaje automático para procesar y generar texto similar al humano basado en grandes cantidades de datos. Están diseñados para entender, interpretar y generar texto de una manera que imita la comunicación humana, aprovechando un preentrenamiento extenso en conjuntos de datos diversos para lograr un alto nivel de comprensión y generación de lenguaje.

Ejemplos de estos modelos son BERT (Google), ELMo (Allen Institute), GPT-3 (OpenAI) y LLaMA (Meta AI), dichos modelos son entrenados en grandes conjuntos de datos de lenguaje natural para predecir la probabilidad de diferentes palabras o frases en un contexto dado. Utilizan redes neuronales profundas para crear representaciones de alta dimensionalidad de las palabras, considerando su contexto y capturando tanto la sintaxis como la semántica [16].

3.1.5.1. Arquitectura Transformer

La arquitectura Transformer, introducida por Vaswani et al. (2017) [17], marcó un punto de inflexión en el desarrollo de modelos de lenguaje natural. A diferencia de arquitecturas recurrentes previas como LSTM o GRU, los Transformers no dependen de una secuencia paso a paso, sino que procesan toda la entrada simultáneamente utilizando mecanismos de atención.

Principio clave: Self-Attention

El componente central del Transformer es el mecanismo de autoatención (*self-attention*), que permite al modelo ponderar la importancia relativa de cada palabra en una secuencia con respecto a las demás. Esto facilita la captura de dependencias contextuales de largo alcance, esenciales para tareas como traducción, generación de texto y clasificación semántica.

Dado un conjunto de vectores de entrada X, la atención se calcula como:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

donde:

- Q (queries), K (keys) y V (values) son matrices derivadas de X.
- d_k es la dimensión de los vectores de atención.

Este mecanismo permite que el modelo determine qué partes del texto son más relevantes para comprender una palabra determinada.

Componentes de un Transformer clásico

Un Transformer consta de bloques encadenados que incluyen:

- **Multi-Head Attention**: Múltiples cabezas de atención permiten al modelo captar distintos tipos de relaciones semánticas.
- **Capa de normalización (LayerNorm)**
- **Capa feed-forward**: Una red neuronal completamente conectada aplicada de forma independiente a cada posición.
- **Residual connections**: Facilitan el entrenamiento de redes profundas evitando el problema de degradación del gradiente.

Ventajas frente a modelos anteriores

- **Paralelización**: Al no requerir procesamiento secuencial como las RNN, los Transformers permiten entrenamiento más rápido.
- **Escalabilidad**: Se pueden entrenar modelos con miles de millones de parámetros sin perder estabilidad.
- **Versatilidad**: La arquitectura se adapta fácilmente a tareas de clasificación, resumen, traducción y generación.

Impacto y evolución

Desde su publicación, la arquitectura Transformer ha sido la base de los modelos más avanzados de PLN, como BERT, GPT, T5 y LLaMA. Su flexibilidad y eficiencia explican su adopción casi universal en la investigación y la industria.

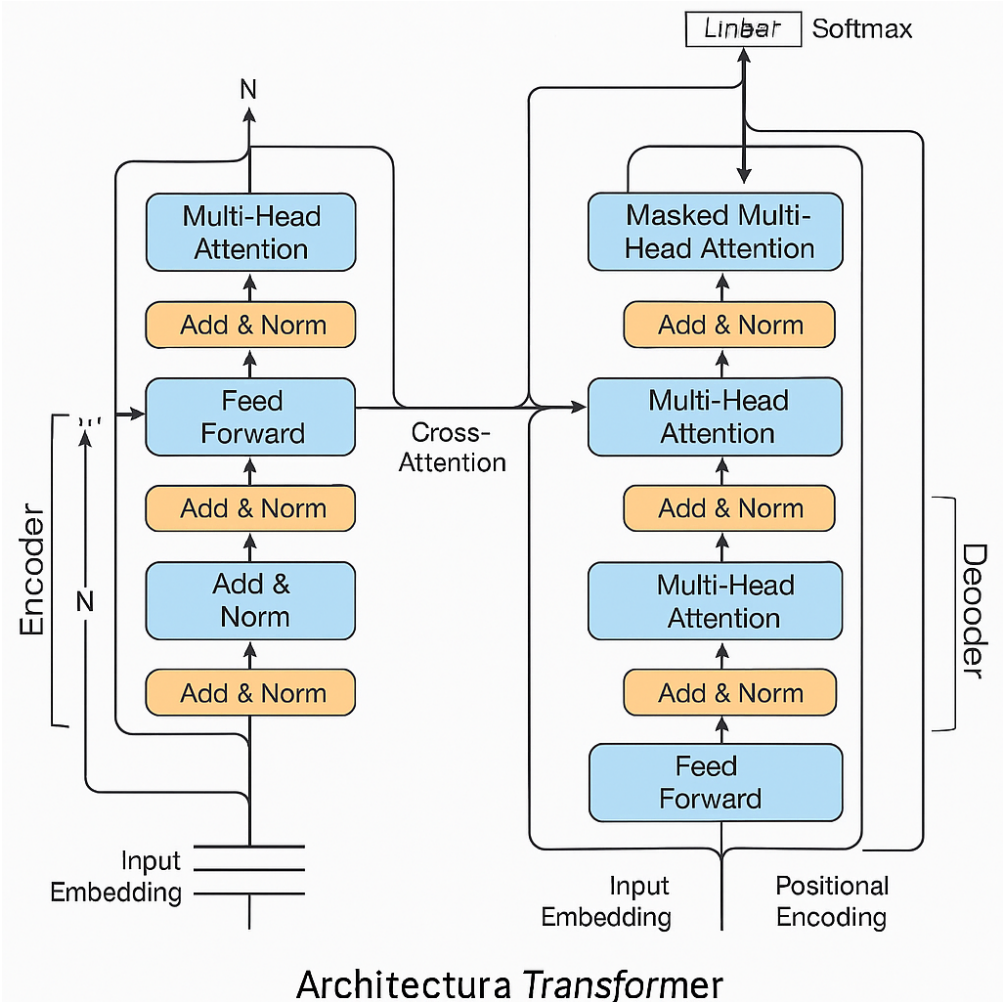


Figura 4.2: Arquitectura Transformer basada en el modelo original de Vaswani et al. (2017)

La figura ilustra 3.2 la arquitectura original del modelo Transformer propuesto por Vaswani et al. (2017), el cual está compuesto por dos bloques principales: el codificador (Encoder) y el decodificador (Decoder). Ambos bloques están construidos a partir de la repetición de múltiples capas idénticas (N capas), y cada una de estas capas contiene componentes clave que permiten al modelo aprender representaciones contextuales complejas.

Resumen del Flujo de Datos

- La entrada textual es transformada en vectores por el `Input Embedding`.
- El encoder procesa esta entrada en paralelo, capturando relaciones contextuales mediante atención múltiple.
- El decodificador genera las salidas palabra por palabra, utilizando tanto la atención propia como la atención cruzada al encoder.

- La salida final es una secuencia generada que representa la predicción del modelo.

3.1.5.2. Fine Tunning (Ajuste finio)

El ajuste fino o fine-tuning en los Modelos de Lenguaje de Gran Escala es el proceso de tomar un modelo previamente entrenado y continuar entrenándolo en un conjunto de datos específico para una tarea particular. Este proceso permite al modelo aprender características y patrones específicos del nuevo conjunto de datos, mejorando su rendimiento en tareas especializadas sin necesidad de entrenar el modelo desde cero. [18] El ajuste fino se puede realizar ajustando algunos de los parámetros del modelo, manteniendo otros congelados, o actualizando todos los parámetros. El objetivo es adaptar el modelo a un dominio específico, lo cual puede ser particularmente útil para aplicaciones que requieren un conocimiento especializado o una comprensión profunda de contextos particulares. [19]

3.1.5.3. Cuantización de Modelos de Lenguaje

3.1.5.3.1. Definición La cuantización es una técnica de optimización que consiste en representar los parámetros de un modelo (como pesos y activaciones) con menor precisión numérica, reemplazando los típicos 32 bits en punto flotante (FP32) por representaciones de menor tamaño, como 8 bits (INT8), 4 bits (INT4) o incluso 2 bits.

Su principal objetivo es reducir significativamente el consumo de memoria y acelerar la inferencia, permitiendo desplegar modelos grandes en dispositivos con recursos limitados, como una sola GPU o incluso hardware embebido. Esta reducción viene acompañada, sin embargo, de un posible deterioro en la precisión, por lo que deben emplearse estrategias que equilibren eficiencia y calidad del modelo.

3.1.5.3.2. Tipos de Cuantización Existen diversos enfoques de cuantización en el aprendizaje profundo, cada uno con ventajas y compromisos distintos:

1. Cuantización Post-Entrenamiento (Post-Training Quantization - PTQ) Consiste en cuantizar un modelo ya entrenado, sin necesidad de realizar un nuevo entrenamiento. Es simple y rápido, pero puede causar degradaciones importantes en la calidad del modelo si no se aplican correcciones.

2. Cuantización Consciente del Entrenamiento (Quantization-Aware Training - QAT) Durante el entrenamiento, se simulan operaciones cuantizadas para que el modelo aprenda a adaptarse a estas restricciones. Este método preserva mejor el rendimiento, pero implica un mayor costo computacional.

3. Cuantización en Inferencia (Runtime Quantization) Aplicada únicamente durante la etapa de inferencia. Puede involucrar conversiones dinámicas (de FP32 a INT8, por ejemplo) o modelos ya entrenados directamente en bajo bit.

3.1.5.4. LoRA: Low-Rank Adaptation

La técnica LoRA (Low-Rank Adaptation) [20] es una solución eficiente para ajustar modelos grandes sin modificar directamente todos sus parámetros. En lugar de actualizar los pesos originales del modelo, LoRA introduce matrices de bajo rango en ciertas capas (por ejemplo, proyecciones de atención como q_proj, k_proj, v_proj), permitiendo que solo estas matrices sean entrenadas.

Esto reduce drásticamente el número de parámetros entrenables, disminuye el uso de memoria y acelera el entrenamiento, manteniendo la calidad del modelo base. LoRA es especialmente útil cuando se requiere entrenar múltiples versiones de un modelo para diferentes tareas o dominios sin duplicar toda su arquitectura.

3.1.5.5. LQLoRA: Quantized Low-Rank Adaptation

QLoRA, propuesto por Dettmers et al. (2023) [21], combina los beneficios de LoRA con la eficiencia de la cuantización en 4 bits. Esta técnica permite entrenar modelos cuantizados —es decir, representados con menor precisión (como INT4 en lugar de FP32)— lo que reduce aún más el uso de memoria y permite entrenar modelos de gran tamaño (como LLaMA2-7B) en una única GPU de 24 GB.

QLoRA conserva la calidad del modelo gracias a: Técnicas como la doble cuantización, Representaciones de baja precisión (nf4), optimizaciones como el gradient checkpointing.

Este enfoque es clave para democratizar el entrenamiento de LLMs, haciendo posible su uso en contextos con infraestructura limitada.

3.1.5.6. Similitud Semántica y Cálculo con la Métrica del Coseno

La **similitud semántica** es una medida del grado de semejanza entre dos expresiones de lenguaje natural, en función de su significado y no únicamente de su forma textual. Esta métrica es esencial en tareas de procesamiento de lenguaje natural (PLN), como recuperación de información, deduplicación de textos, búsqueda semántica y análisis de similitud entre oraciones.[22]

Para cuantificar esta similitud, los textos se transforman en vectores en un espacio semántico de alta dimensión mediante modelos de embeddings (por ejemplo, `Sentence-BERT`).[23]Luego, se compara la orientación de estos vectores utilizando la **similitud del coseno**, una métrica ampliamente utilizada en PLN y recuperación de información.

Similitud del Coseno

La similitud del coseno mide el ángulo entre dos vectores $\vec{A}$ y $\vec{B}$ en un espacio n-dimensional, y se define como:

$$\text{sim}_{\cos}(\vec{A}, \vec{B}) = \frac{\vec{A} \cdot \vec{B}}{\|\vec{A}\| \cdot \|\vec{B}\|}$$

donde:

- $\vec{A} \cdot \vec{B}$ es el producto punto entre los vectores,
- $\|\vec{A}\|$ y $\|\vec{B}\|$ son las normas (magnitudes) de los vectores.

Esta métrica produce valores en el rango de $[-1, 1]$:

- Un valor de **1** indica que los vectores son idénticos en dirección (alta similitud).
- Un valor de **0** indica que son ortogonales (sin relación semántica).
- Un valor de **-1** indicaría direcciones opuestas (poco común en embeddings semánticos).

Ventajas de la Similitud del Coseno

- Es **invariante a la magnitud**, por lo que se enfoca solo en la orientación del vector.
- Es adecuada para comparar textos representados mediante embeddings, donde la semántica se encuentra en la dirección del vector y no en su longitud.
- Su cálculo es eficiente computacionalmente y escalable a grandes volúmenes de datos.

En el presente trabajo, se utilizó la similitud del coseno para aplicar **deduplicación semántica**, evaluando la proximidad entre vectores generados por el modelo `Sentence-BERT`, lo que permitió eliminar registros redundantes del conjunto de datos de entrenamiento.

3.1.5.7. FAISS: Búsqueda de Vectores a Gran Escala

Para acelerar el proceso de comparación entre miles de embeddings, se empleó FAISS (Facebook AI Similarity Search), una librería eficiente para búsquedas de similitud entre vectores densos [24]. FAISS permite realizar búsquedas de vecinos más cercanos (nearest neighbors) a alta velocidad, incluso en grandes volúmenes de datos, gracias a su optimización con GPU y estructuras de indexación avanzadas.

Esta herramienta fue clave para aplicar deduplicación semántica en el dataset previo al entrenamiento.

3.1.5.8. Evaluación de Modelos con RAGAS

Para evaluar la calidad de los resultados generados por el modelo, se utilizó el framework RAGAS (Retrieval-Augmented Generation Assessment Score) [25]. Este sistema permite evaluar modelos generativos considerando no solo métricas tradicionales (como BLEU o ROUGE), sino también aspectos más modernos como:

- **Faithfulness**: Evalúa si las respuestas generadas son fieles al contexto proporcionado.
- **Answer Relevancy**: Mide la relevancia de la respuesta con respecto a la intención del usuario.
- **Context Precision**: Determina si la información usada proviene efectivamente del contexto dado.

Estas métricas son fundamentales para asegurar que el modelo sea útil en aplicaciones sensibles como la atención al cliente, donde la precisión y coherencia son esenciales.

4.1.6. Soluciones Agénticas para Atención al Cliente

¿Qué es un agente conversacional?

Un agente conversacional, o chatbot, es un programa automatizado que utiliza técnicas de procesamiento de lenguaje natural y aprendizaje automático para interactuar con los usuarios en lenguaje natural. Su objetivo principal es automatizar tareas de atención al cliente, tales como responder preguntas frecuentes, generar tickets o dirigir solicitudes específicas, aportando escalabilidad, disponibilidad 24/7 y reducción de carga para los agentes humanos [26], [27].

Tipos de Agentes en Atención al Cliente

Según la literatura reciente, los agentes pueden clasificarse así **hussein2019survey**, [26]:

- **Agentes reactivos**: Responden con respuestas predefinidas basadas en patrones, sin gestionar el contexto conversacional.
- **Agentes conversacionales**: Mantienen el diálogo mediante modelos de lenguaje y gestores de diálogo, integrando contexto en la interacción.
- **Agentes autónomos ("agentic")**: Realizan múltiples acciones, manejan razonamiento y pueden tomar iniciativas como escalar a un agente humano.
- **Agentes híbridos humano-IA**: Combinan automatización y supervisión humana, activándose solo en escenarios complejos.

Casos de Uso en CRM

1. Plataforma IBM Watson Assistant en Banca IBM Watson Assistant ha sido utilizado en bancos para automatizar consultas frecuentes, seguimiento de transacciones y apertura de tickets, logrando una atención más rápida y consistente, reduciendo costos operativos y aumentando la satisfacción del cliente [28].

2. CRMArena: Evaluación de agentes LLMs El benchmark CRMArena evalúa agentes generativos en nueve tareas reales de CRM —como generación de tickets y seguimiento de reembolsos— obteniendo puntajes entre 38 % y 55 %, lo que revela tanto el potencial como las limitaciones actuales en ambientes CRM [29].

3. Análisis comparativo de chatbots empresariales Un estudio de Ehsani et al. (2023) comparó chatbots de atención al cliente en diversos sectores (retail, telecomunicaciones, banca). Se concluyó que la clave está en la eficiencia de recuperación de información y usabilidad, especialmente en CRM donde la precisión y agilidad son críticas [27].

4.2. Antecedentes

Adicional a los conceptos mencionados previamente, a continuación, se ofrece una visión general de los artículos encontrados en la literatura relacionados directamente con el proyecto en cuestión. El propósito es identificar antecedentes que aborden problemas similares y establecer así una base teórica sólida para el proyecto.

La búsqueda se llevó a cabo en las bases de datos científicas IEEE y ScienceDirect, que cubren ampliamente áreas de conocimiento cruciales para la ciencia de datos, utilizando los conceptos previamente descritos como palabras clave para una búsqueda más precisa y relevante.

Se seleccionaron los cinco artículos más relevantes, considerando el orden de relevancia proporcionado por la base de datos y la conexión con el proyecto, de tal manera que se abordarán los cinco pilares del proyecto a saber: integración y construcción de IA con datos de CRM, entendimiento del cliente a través de los datos almacenados en un CRM, modelos LLM, modelos de Análisis de Sentimientos y la creación de un ChatBot que utilice IA para interactuar con los clientes. A continuación, se proporciona una breve descripción de estos artículos y se detalla su relación con el proyecto.

4.2.1. Integración y construcción de IA con datos de CRM

Nombre del artículo: Integration of AI in CRM: Challenges and guidelines

3.2.1.1. Descripción

Este estudio examina la integración de la inteligencia artificial en la Gestión de Relaciones con Clientes CRM, identificando once desafíos específicos a lo largo de cuatro fases de implementación. A través de entrevistas cualitativas con diversos actores, se proporcionan pautas para abordar eficazmente estos desafíos. Al comparar casos exitosos y fallidos, se ofrece una comprensión empírica sobre cómo

diseñar, implementar y mantener proyectos de inteligencia artificial en CRM, subrayando una perspectiva a largo plazo en la utilización de la IA para las relaciones con los clientes. Los hallazgos buscan profundizar la comprensión de las actividades y capacidades necesarias para superar los obstáculos en esta integración, proporcionando preguntas esenciales para los gerentes que emprenden este viaje, en última instancia, ofreciendo una guía valiosa para las empresas en este proceso [9].

3.2.1.2. Conexión con el proyecto

El artículo descrito previamente proporciona una valiosa perspectiva para el proyecto que se desarrolló, ya que proporciona una comprensión detallada de los desafíos y oportunidades en la implementación de tecnologías avanzadas, como lo es el Procesamiento del Lenguaje Natural, esto en el contexto del CRM. Esta investigación ofrece pautas y lecciones aprendidas que fueron relevantes para el desarrollo e implementación del proyecto, permitiendo así alcanzar los objetivos de este.

4.2.2. Entendimiento del cliente a través de los datos de un CRM

Nombre del artículo: A Social CRM Analytic Framework for Improving Customer Retention, Acquisition, and Conversion.

3.2.2.1. Descripción

El artículo presenta un marco analítico de CRM social destinado a mejorar la retención, adquisición y conversión de clientes. Este marco aborda los desafíos de integrar datos de redes sociales en sistemas CRM mediante enfoques avanzados de análisis, demostrando su eficacia en la extracción de información relevante y apoyo a la toma de decisiones. Se estructura en tres módulos principales: clasificación de requisitos del cliente, clasificación y agrupación de clientes, y mejora de la adquisición, retención y conversión de clientes. Los resultados muestran altos niveles de precisión y exactitud en la clasificación de requisitos y sentimientos del cliente, así como la eficiencia del modelo de clasificación de clientes con Random Forest. Las implicaciones para la práctica y la investigación destacan la importancia de mantener el compromiso con los clientes a lo largo de su ciclo de vida, así como la necesidad de datos reales para mejorar la actividad de comunicación en redes sociales [8].

3.2.2.2. Conexión con el proyecto

El artículo descrito proporciona un enfoque integral de gestión de relaciones con clientes (CRM), utilizando técnicas de ciencia de datos en la clasificación de requisitos y sentimientos del cliente, como el modelo Random Forest. Además, ofrece una amplia literatura sobre el entendimiento del cliente en el contexto de

los CRM. Esta información y técnicas representan insumos fundamentales para el desarrollo del proyecto, no solo para comprender al cliente, sino también para materializar técnicas que permitan dicho entendimiento a través del aprovechamiento de los datos en el contexto de un CRM.

4.2.3. Modelos LLM

Nombre del artículo: The Recent Large Language Models in NLP

3.2.3.1. Descripción

Este artículo ofrece una visión general de la evolución reciente en el campo del Procesamiento del Lenguaje Natural (NLP) gracias al desarrollo de Modelos de Lenguaje Grandes (LLM). En particular, se enfoca en cuatro modelos destacados: BERT de Google, ELMo del Instituto Allen, GPT-3 de OpenAI y LLaMA de Meta AI. Para cada uno de estos modelos, se analiza su arquitectura, los conjuntos de datos en los que fueron entrenados, su evaluación de rendimiento, así como sus fortalezas y desafíos. El objetivo del estudio es comparar estos modelos y sus contribuciones al campo del NLP, destacando su impacto en tareas como respuesta a preguntas, análisis de sentimientos y generación de texto [16].

3.2.3.2. Conexión con el proyecto

El artículo descrito anteriormente muestra una visión detallada de LLM avanzados, tales como BERT, ELMo, GPT-3 y LLaMA destacando su capacidad para transformar la interacción entre las máquinas y el lenguaje humano y analizando su arquitectura, conjuntos de datos de entrenamiento, evaluación de rendimiento y desafíos; proporcionando así una base sólida para implementar tecnologías avanzadas en el análisis de datos del cliente haciendo uso de herramientas del NLP como lo es el LLM, la cual fue usada en el proyecto como previamente se mencionó.

4.2.4. Modelos de Análisis de Sentimientos

Nombre del artículo: Sentiment Analysis Based on Deep Learning Approaches.

3.2.4.1. Descripción

El artículo aborda la importancia del análisis de sentimientos para gestionar grandes cantidades de datos generados en redes sociales, foros, blogs, etc., en forma de opiniones y emociones de los usuarios. Se destaca la integración de algoritmos de aprendizaje automático y aprendizaje profundo para mejorar la efectividad del análisis de sentimientos, lo que lo convierte en una herramienta ampliamente utilizada en diversas industrias. En la conclusión, se resalta la relevancia del análisis de sentimientos para entender la opinión de los clientes, mejorar productos y enriquecer la experiencia del usuario, con aplicaciones específicas en redes sociales,

política y comercio electrónico, además de revisar técnicas de aprendizaje profundo y proponer un método basado en redes neuronales recurrentes para el análisis de sentimientos [30].

Adicionalmente, el artículo proporciona una revisión sobre cómo los investigadores emplean técnicas de aprendizaje profundo en diversas aplicaciones del análisis de sentimientos. Detalla varias técnicas basadas en el aprendizaje profundo para llevar a cabo este análisis de sentimientos de manera efectiva, destacando así los estudios más recientes en este campo [30].

3.2.4.2. Conexión con el proyecto

En relación con el proyecto, este artículo proporciona una base sólida al evidenciar el uso del análisis de sentimientos a través de técnicas de aprendizaje profundo. Esta técnica sirvió como referencia para entender como se usa el análisis de sentimientos para analizar de manera inteligente y automatizada los datos del cliente, identificando tendencias, preferencias y puntos de fricción que puedan afectar su satisfacción y lealtad hacia la marca. La evaluación de la efectividad de estos modelos mediante métricas adecuadas contribuyo a determinar la técnica más idónea, asegurando así resultados óptimos y cumpliendo con los objetivos establecidos en el proyecto.

4.2.5. Creación de ChatBot que utilice IA

Nombre del artículo: Interactive Applied Graph Chatbot with Semantic Recognition.

3.2.5.1. Descripción

El artículo presenta un estudio sobre el desarrollo de un chatbot para empresas, destacando tres características principales. Primero, emplea un análisis afectivo del sentimiento, inspirado en el Modelo Circunplexo de Russell, para comprender la polaridad y el grado de afecto en las interacciones de los usuarios. Segundo, ofrece un análisis del sentimiento en frases con conjunciones, desglosando las oraciones según las conjunciones y aplicando análisis afectivo a cada parte para determinar el sentimiento completo de la oración. Por último, introduce un Chatbot de Grafo que utiliza un mapa de conversación visual desarrollado con Vue.js, garantizando una interacción efectiva y una precisión del 100 % en el manejo de conversaciones procedurales [31].

En conclusión, el estudio propone una aplicación generadora de chatbots fácil de usar, especialmente resaltando la capacidad del Chatbot de Grafo para su adaptabilidad por parte de no programadores [31].

3.2.5.2. Conexión con el proyecto

Como se muestra en la descripción del artículo, este se centra en el desarrollo de un ChatBot y técnicas de análisis de sentimientos utilizando inteligencia artificial e ingeniería de software, proporcionando así herramientas valiosas que pueden ser adaptadas para mejorar la interacción con los clientes y, por ende, su satisfacción. Las técnicas presentadas, como el análisis de sentimientos y la implementación de un ChatBot pueden ser integradas en un marco más amplio de análisis de datos y atención al cliente, lo que contribuyó a alcanzar los objetivos del proyecto.

Capítulo 5

Dataset Utilizado

El conjunto de datos utilizado en este proyecto fue empleado tanto para la creación del modelo de análisis de sentimientos como para el proceso de *fine-tuning* de LLMs. Este dataset, disponible públicamente en la plataforma *Hugging Face*, contiene **26.872 interacciones textuales** entre usuarios y un sistema de atención al cliente (CRM). Cada registro está compuesto por una **entrada del usuario** (consulta o instrucción) y la **respuesta generada por el sistema**.

Con el objetivo de enriquecer esta base de datos, se extrajo una muestra de **1.011 registros**, la cual fue clasificada manualmente según el **sentimiento predominante** expresado por el usuario (ver detalle en el Capítulo 6). Esta muestra se utilizó para entrenar un modelo de análisis de sentimientos. Posteriormente, dicho modelo fue aplicado al conjunto completo, generando así una tercera variable: el sentimiento del usuario. Esta variable adicional fue clave para mejorar la empatía y adecuación de las respuestas generadas por los modelos ajustados mediante *fine-tuning*.

5.1. Descripción del Dataset Utilizado

El dataset general presenta una diversidad temática estructurada en **12 categorías** y **27 intenciones**, cubriendo escenarios frecuentes en entornos de servicio al cliente digital, como gestión de pedidos, pagos, atención a problemas de cuenta, contacto con agentes, entre otros. Las características principales del dataset se resumen a continuación:

- **Número de registros**: 26.872 pares de interacción reales.
- **Número de categorías**: 12 áreas funcionales (pedidos, pagos, cuenta, contacto, etc.).
- **Número de intenciones**: 27 acciones específicas asociadas a las categorías.

Categoría	Intenciones
Pedido	Realizar pedido, Cambiar pedido, Cancelar pedido, Consultar cargo por cancelación
Envío / Entrega	Opciones de entrega, Tiempo estimado de entrega, Rastrear pedido, Establecer dirección de envío, Cambiar dirección de envío
Pago	Consultar métodos de pago, Problema con el pago
Factura	Consultar factura, Obtener factura
Reembolso	Obtener reembolso, Consultar política de reembolso, Rastrear reembolso
Cuenta	Crear cuenta, Eliminar cuenta, Editar cuenta, Cambiar de cuenta, Recuperar contraseña, Problemas de registro
Contacto	Contactar servicio al cliente, Contactar agente humano
Retroalimentación	Presentar una queja, Dejar reseña
Suscripción	Suscribirse al boletín

Cuadro 5.1: Agrupación de intenciones según la categoría del dataset

Esta riqueza temática constituye una base robusta y representativa para entrenar modelos de lenguaje natural capaces de gestionar una amplia variedad de solicitudes, facilitando la personalización de respuestas y mejorando significativamente la experiencia del usuario.

5.2. Delimitación del Dominio de Aplicación

Con el fin de garantizar la coherencia, pertinencia y efectividad de los modelos desarrollados, se estableció un dominio de aplicación claramente delimitado: **interacciones de atención al cliente en servicios digitales de e-commerce y tecnología, con enfoque en el ámbito hispanohablante**.

El dataset empleado, identificado como **bitext-customer-support-llm-chatbot-training-dataset**, se encuentra disponible públicamente en la plataforma *Hugging Face* dando clic **aquí**. Este conjunto de datos fue recopilado de un sistema CRM operativo en contextos reales, lo que garantiza que los patrones lingüísticos, el contenido temático y las necesidades expresadas reflejan con fidelidad los escenarios comunes del sector.

Todas las etapas del proyecto —incluyendo el análisis exploratorio, la construcción del modelo de análisis de sentimientos, la generación de una nueva variable emocional y el proceso de *fine-tuning*— se realizaron considerando exclusivamente este dominio específico. Esta estrategia fue clave para aumentar la relevancia contextual de las respuestas generadas, así como para reducir el riesgo de *alucinaciones*, es decir, respuestas incorrectas o fuera de contexto al aplicar el modelo en

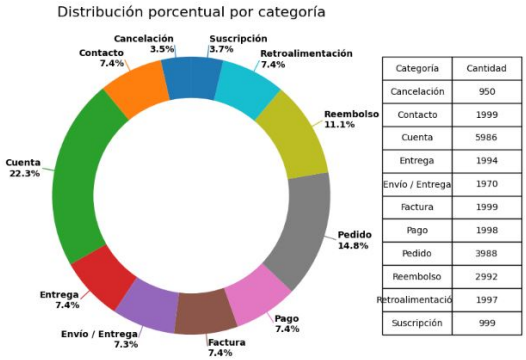
escenarios para los que no fue entrenado.

Por tanto, aunque el modelo demuestra un alto desempeño en el contexto definido, se recomienda cautela en su despliegue en otros sectores, dada la sensibilidad de los modelos de lenguaje a las características semánticas del dominio sobre el cual fueron ajustados.

5.3. Análisis Exploratorio

El análisis exploratorio de datos (EDA, por sus siglas en inglés) fue un paso inicial crucial para comprender la estructura y calidad del dataset. A través de diversas técnicas y visualizaciones, se logró obtener información valiosa que guiaría las fases posteriores del proyecto. Los principales hallazgos encontrados durante el EDA fueron:

5.3.1. Distribución porcentual por categoría e intención



Categoria	Cantidad
Cancelación	950
Contacto	1999
Cuenta	5986
Entrega	1994
Envío / Entrega	1970
Factura	1999
Pago	1998
Pedido	3988
Reembolso	2992
Retroalimentació	1997
Suscripción	999

Figura 5.1: Distribución de registros por categoría

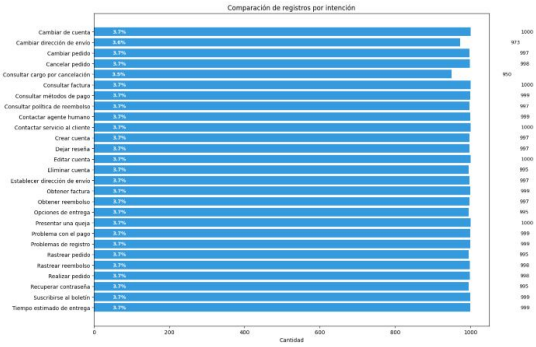


Figura 5.2: Distribución de registros por intención

Análisis y conclusión: Las gráficas presentadas permiten observar que el dataset cuenta con una distribución equilibrada tanto en las categorías como en las intenciones de los usuarios. En la primera imagen, se destacan categorías como *Contactar servicio al cliente*, *Cambiar de cuenta* y *Presentar una queja*, todas con un volumen significativo de registros, cercano a los 1000, lo cual evidencia una cobertura representativa de los escenarios más comunes en la interacción con servicios digitales. En la segunda gráfica, de tipo anillo, se aprecia que las intenciones están bien distribuidas, sin una dominancia excesiva de alguna en particular; si bien sobresalen levemente categorías como *Operaciones de cuenta* y *Consultas generales*, el conjunto mantiene un balance adecuado que reduce el riesgo de sesgo en tareas de clasificación. En conjunto, esta diversidad garantiza una base sólida para el entrenamiento de modelos robustos y generalizables, permitiendo una aplicación efectiva en contextos reales de atención al cliente.

5.3.2. Distribución del Número de Caracteres

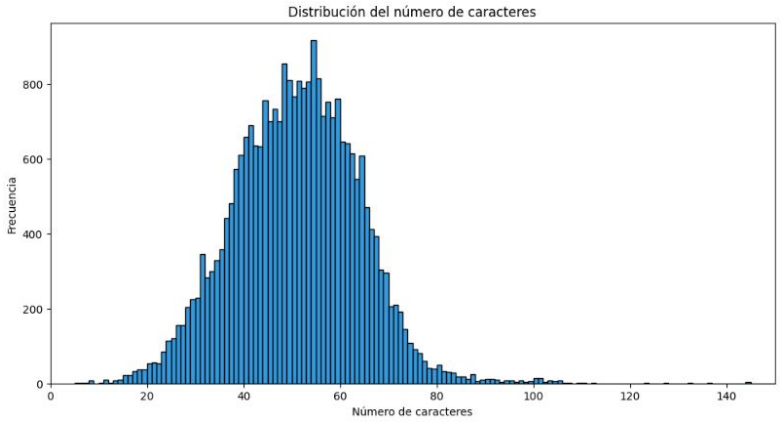


Figura 5.3: Distribución del número de caracteres en los registros

Análisis y conclusión: El gráfico de distribución del número de caracteres en los registros muestra que la mayoría de los textos se concentran en un rango medio, entre aproximadamente **20** y **80** caracteres, con una tendencia clara hacia valores alrededor de los **50** a **60** caracteres. Se observa una distribución relativamente amplia, con cantidades decrecientes a medida que los registros son más largos o más cortos que ese rango central. Hay picos notables en las longitudes cercanas a **44**, **48** y **54** caracteres, indicando que varios registros comparten tamaños similares. También se aprecian registros con longitudes mucho mayores, aunque en cantidades muy reducidas, que podrían ser casos especiales o outliers. En general, la distribución es bastante dispersa pero con concentración en un rango intermedio, lo que es habitual en datos textuales con variabilidad moderada en la longitud de los registros.

5.3.3. Distribución de Tokens

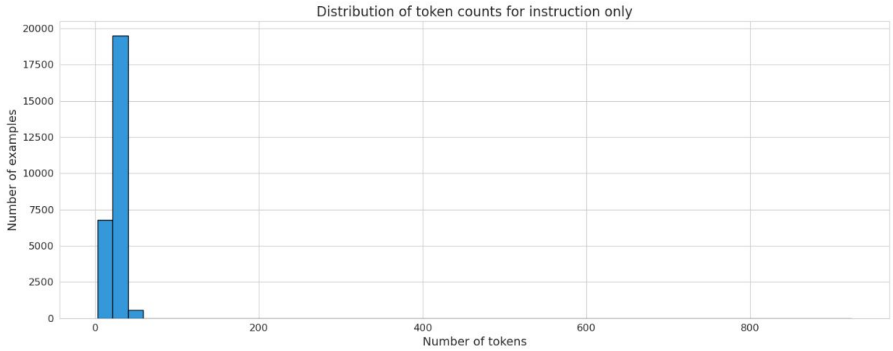


Figura 5.4: Distribución de recuentos de tokens para instrucciones antes de la limpieza.

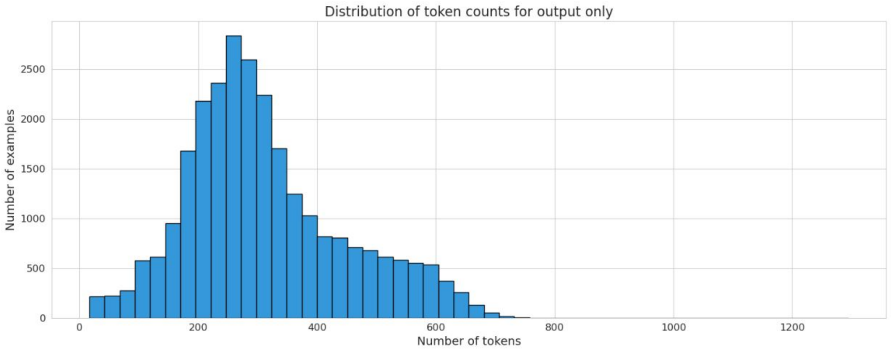


Figura 5.5: Distribución de recuentos de tokens para salidas antes de la limpieza.

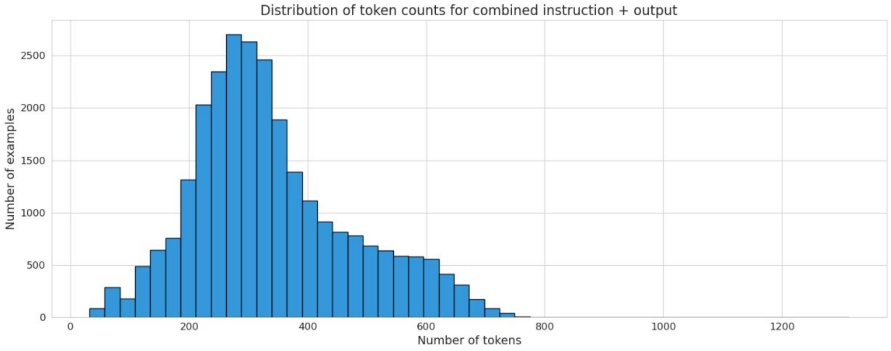


Figura 5.6: Distribución de recuentos de tokens para instrucciones y salidas combinadas antes de la limpieza.

Análisis y conclusión: En las imágenes anteriores se puede observar que la distribución de recuentos de tokens para instrucciones muestra una alta concentración de registros en el rango de 0 a 10 tokens, con aproximadamente 4,025 muestras, y un notable incremento en el rango de 20 a 30 tokens, alcanzando cerca de 12,025 registros. Por otro lado, la representación de las salidas refleja una distribución simétrica similar a la normal alrededor de 280 tokens, sugiriendo una consistencia en las respuestas generadas. Esta tendencia se reafirma en la distribución combinada de instrucciones y salidas, que también muestra simetría alrededor de los 280 tokens. Estos patrones iniciales proporcionan una base para orientar el proceso de limpieza y ajuste del dataset, asegurando que el conjunto de datos sea más uniforme y adecuado para el entrenamiento de modelos.

Capítulo 6

Comprensión Emocional del Cliente a través de Modelos de Análisis de Sentimientos

Como se planteó inicialmente uno de los objetivos propuestos, es el uso de modelos de análisis de sentimientos, de tal manera que estos sirvan para identificar el estado emocional del cliente en cada interacción con el sistema y de esa manera customizar la respuesta que el sistema le proporcionará a dicho cliente. Para cumplir con este propósito, se desarrollo un modelo de análisis de sentimientos capaz de clasificar las interacciones de los clientes en tres categorías: **positivo**, **negativo** y **neutro**. A continuación, se presenta de manera detallada todo el proceso seguido para el desarrollo del modelo, el cual incluyó las siguientes etapas: recolección, exploración,limpieza de y preprocesamiento de los los datos; así como la selección, entrenamiento, evaluación del modelo y análisis de los resultados obtenidos.

6.1. Limpieza y Preprocesamiento

6.1.1. Descripción del Dataset Inicial

El dataset utilizado para entrenar el modelo de análisis de sentimientos corresponde a una **muestra de 1.011 registros** extraída del conjunto de datos general descrito previamente en el Capítulo 5. Esta muestra fue construida a partir de interacciones reales entre usuarios y un sistema de atención al cliente (CRM), y cada registro fue **clasificado manualmente** según el sentimiento predominante expresado en la instrucción del usuario.

La finalidad de esta muestra etiquetada fue servir como base para el entrenamiento supervisado del modelo de análisis de sentimientos. Una vez entrenado y validado, el modelo fue utilizado para predecir el sentimiento del resto de interacciones presentes en la base de datos general. Esta etapa permitió enriquecer cada registro con una tercera variable —el sentimiento del usuario— y habilitó la posterior integración de esta información en modelos de lenguaje ajustados mediante

fine-tuning. Para una descripción detallada del origen y estructura del dataset completo, véase el Capítulo 5.

6.1.2. Limpieza de los Datos

- **Eliminación de textos muy cortos**: Se excluyeron **674 registros** cuyo número de caracteres era **igual o inferior a 25**, ya que estos textos no contenían suficiente información semántica. Esta medida, común en proyectos de procesamiento de lenguaje natural (PLN), contribuyó a eliminar entradas ruidosas y no representativas del fenómeno que se quiere modelar.
- **Eliminación de registros duplicados**: Durante el análisis exploratorio del conjunto de datos, se identificaron **8,168 registros duplicados** en la variable correspondiente a las entradas del usuario. Esta redundancia representaba un riesgo importante de sobreajuste y sesgo durante el entrenamiento del modelo, por esta razón, dichos registros se eliminaron reduciendo asi el conjunto original de **26,872 registros** a **18,704 registros únicos**.
- **Conversión a minúsculas**: Se transformaron todos los textos a minúsculas para garantizar uniformidad en el análisis textual, evitando que palabras iguales con diferentes capitalizaciones se consideren distintas.
- **Normalización del uso de espacios**: Se corrigió el uso inadecuado de espacios en blanco, eliminando espacios adicionales y ajustando los espacios antes de los signos de puntuación. Esta normalización contribuyó a una segmentación más precisa y a una mayor coherencia en el formato textual.
- **Corrección de errores tipográficos y eliminación de caracteres especiales**: Se revisaron los textos para corregir errores ortográficos frecuentes y eliminar caracteres no alfabéticos innecesarios. Esta etapa mejoró la calidad léxica de los datos y aseguró un formato uniforme para el procesamiento posterior.

6.1.3. Construcción del Dataset para el Modelado

Con el fin de construir un conjunto de datos representativo, equilibrado y etiquetado de manera confiable para entrenar los modelos de análisis de sentimientos, se implementó una estrategia en varias etapas. Esta estrategia consistió en: (1) aplicar una clasificación inicial automática a la población completa utilizando un modelo preentrenado, (2) realizar un muestreo aleatorio estratificado para seleccionar una muestra balanceada por clase, y (3) efectuar una clasificación manual cuidadosa de esa muestra utilizando criterios lingüísticos y contextuales definidos.

6.4.1 Estrategia Utilizada

La estrategia aplicada para construir el dataset se diseñó con el objetivo de optimizar la cobertura y representatividad del conjunto de datos de entrenamiento.

A continuación se resumen sus fases:

1. **Muestreo aleatorio estratificado:** A partir de la clasificación automática, se extrajo una muestra proporcional por cada clase, con el fin de construir un conjunto de entrenamiento balanceado.

2. **Clasificación manual de la muestra:** La muestra extraída fue posteriormente etiquetada de forma manual, siguiendo criterios previamente definidos para asegurar la calidad de las etiquetas utilizadas en el entrenamiento.

Este enfoque mixto permite beneficiarse de la rapidez del modelo automático sin sacrificar la calidad del etiquetado gracias al componente humano final.

6.4.2 Clasificación Automática con Modelo Preentrenado

Para la clasificación inicial, se utilizó el modelo `Startup-Exchange/tps_sentimental_analys` disponible en la plataforma Hugging Face. Este modelo, basado en la arquitectura `BERT`, ha sido preentrenado sobre un corpus diverso de reseñas de productos y servicios, y optimizado para detectar el tono de los textos, categorizándolos en tres clases: **Positive**, **Neutral** y **Negative**

Características del modelo:

- **Arquitectura:** BERT
- **Corpus de entrenamiento:** Reseñas mixtas de comercio electrónico, servicios y atención al cliente
- **Métricas de desempeño reportadas por los autores:**
  - Precisión (Accuracy): **87.4 %**
  - F1-Score promedio: **0.86**
  - F1-Score por clase: Positive (0.88), Neutral (0.83), Negative (0.87)
- **Plataforma de implementación:** Hugging Face Transformers Pipeline

El modelo fue implementado mediante un pipeline de inferencia, el cual procesó automáticamente cada uno de los registros del conjunto de datos previamente depurado. El proceso de clasificación tuvo una duración aproximada de **49.21 minutos**. Los resultados obtenidos fueron almacenados en un nuevo archivo con la información enriquecida con las etiquetas de sentimiento correspondientes.

6.4.3 Muestreo Aleatorio Estratificado

Para construir una muestra de entrenamiento balanceada, se calculó el tamaño de muestra necesario utilizando la fórmula para poblaciones finitas, con los siguientes parámetros:

- Nivel de confianza: 95 % ($Z = 1{,}96$)
- Proporción esperada: $p = 0{,}5$
- Margen de error: $e = 0{,}03$
- Tamaño poblacional: 26,872 registros

El tamaño de muestra calculado fue de **1,011 registros**. Sin embargo, durante la clasificación automática se evidenció una distribución desbalanceada entre las clases (Ver Figura 6.1), destacando un número particularmente bajo de registros con sentimiento *Positive* (solo 51 instancias en toda la base). Para corregir este desequilibrio y permitir un muestreo estratificado balanceado, se generaron artificialmente **360 registros positivos** adicionales mediante técnicas de **generación sintética de datos**. Estos textos fueron creados con el modelo de lenguaje **GPT-4-turbo**, desarrollado por OpenAI, el cual recibió instrucciones específicas mediante prompts diseñados para producir ejemplos coherentes y representativos del sentimiento positivo, preservando la semántica y la estructura observada en los textos originales del conjunto de datos.

Posteriormente, se aplicó un muestreo aleatorio estratificado en función de las etiquetas de sentimiento, extrayendo un número equitativo por clase. A continuación, se muestran los resultados obtenidos.

6.4.4 Distribución de la Muestra

El conjunto de datos de la muestra quedó conformado por un total de **1,011 registros**, distribuidos equitativamente entre las tres clases de sentimiento: *Positive* (337), *Neutral* (337) y *Negative* (337). Esta distribución balanceada se logró mediante el muestreo aleatorio estratificado explicado previamente. A continuación, se muestra la distribución de las categorias en la población Vs la muestra.

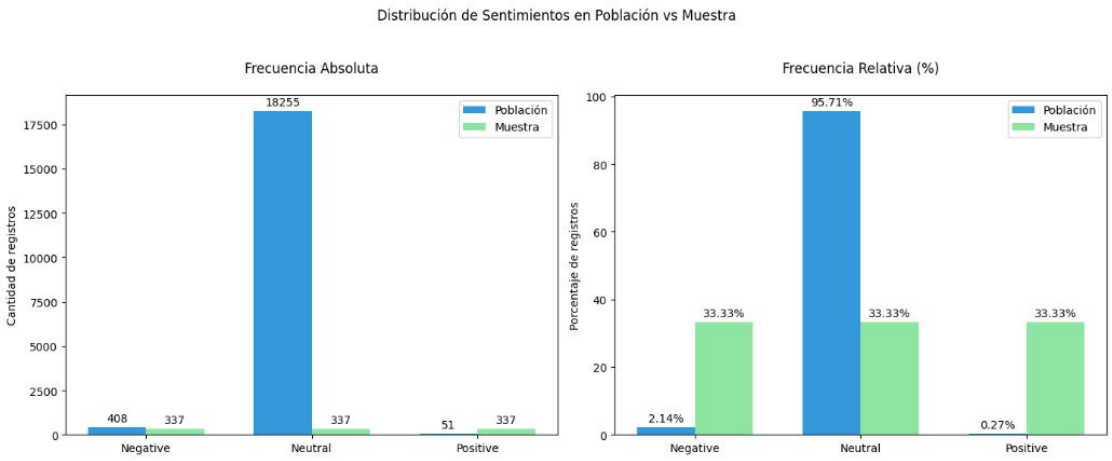


Figura 6.1: Distribución de la población original versus la muestra balanceada utilizada para entrenamiento.

Análisis y conclusión: La gráfica evidencia un marcado desbalance en la población original, donde la clase *Neutral* representa más del 95 % de los registros, dejando una representación mínima a las clases *Positive* y *Negative*. En contraste, la muestra construida presenta una distribución perfectamente equitativa entre las tres clases, lo que es fundamental para entrenar modelos de análisis de sentimientos robustos y no sesgados. Este balance cobra aún más relevancia al considerar que se dispone de un conjunto reducido de registros ($n = 1011$), ya que mejora la capacidad del modelo para aprender a distinguir adecuadamente cada polaridad, maximizando su poder predictivo y reduciendo el riesgo de sobreajuste a la clase mayoritaria.

6.4.5 Clasificación Manual Basada en Criterios Lingüísticos

Finalmente, la muestra de 1,011 registros fue sometida a una revisión y clasificación manual, considerando criterios lingüísticos específicos para garantizar un etiquetado de alta calidad. Cada reseña fue analizada en función del tono general, expresiones emocionales, y contenido sobre el producto o servicio.

Criterios utilizados:

- **Sentimiento Positivo:**
  - Expresiones de satisfacción o alegría.
  - Recomendaciones explícitas.
  - Valoraciones positivas del servicio al cliente.
  - Reconocimiento de alta calidad del producto o servicio.
- **Sentimiento Negativo:**
  - Expresiones de frustración, queja o insatisfacción.
  - Mención de problemas logísticos (envíos, entregas).
  - Críticas a la atención al cliente o al funcionamiento del producto.
- **Sentimiento Neutral:**
  - Comentarios informativos sin carga emocional.
  - Preguntas o afirmaciones sin juicio de valor.
  - Observaciones objetivas o solicitudes sin emociones explícitas.

Ejemplos ilustrativos:

- **Positivo:** *"¡Me encanta este producto! Superó todas mis expectativas y sin duda lo recomendaría a mis amigos y familiares."*
  Razón: expresa entusiasmo, satisfacción completa y una recomendación explícita.

- **Negativo:** *"¡El pedido llegó tarde y en muy mal estado! estoy muy decepcionado con el producto y muy ofendido con el servicio recibido."*
  Razón: expresa insatisfacción, decepción y crítica negativa clara hacia el servicio.

- **Neutral:** *"¿Cómo puedo consultar las opciones de pago disponibles?"*
  Razón: se trata de una consulta informativa sin expresión emocional evidente.

Este proceso manual permitió corregir posibles errores del modelo preentrenado utilizado y refinar el conjunto de entrenamiento, asegurando mayor coherencia y calidad para el desarrollo posterior de modelos de análisis de sentimientos.

6.4.6 Dataset Creado

Una vez realizada la clasificación manual de los 1,011 registros que conforman la muestra, se generó el conjunto de datos definitivo que fue utilizado para el entrenamiento de los modelos de análisis de sentimientos. A continuación, se presenta una comparación gráfica entre la distribución obtenida con el modelo preentrenado y la distribución final luego de la revisión manual.

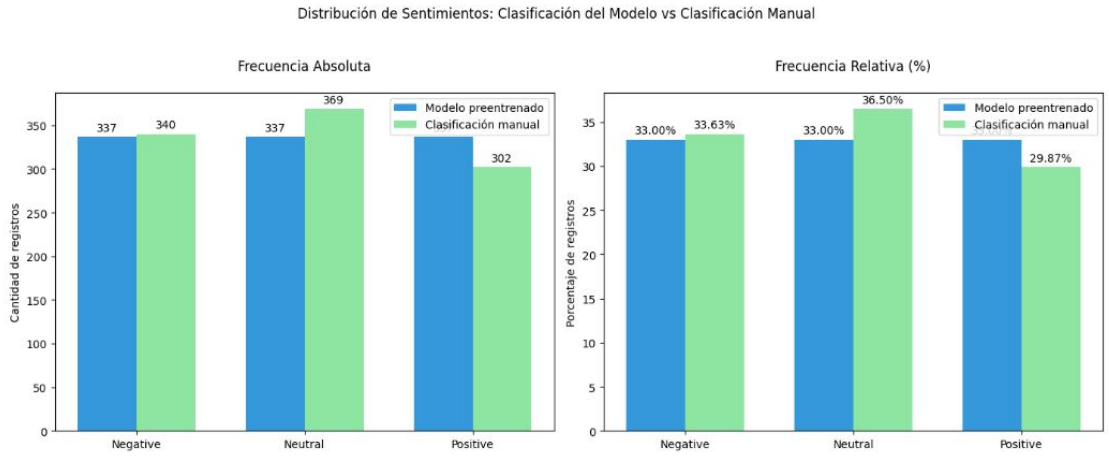


Figura 6.2: Distribución de la clasificación automática del modelo preentrenado versus la clasificación manual definitiva.

Análisis y conclusión: La anterior grafica muestra que, si bien el modelo preentrenado generó una distribución perfectamente balanceada entre las tres clases de sentimiento (33 % cada una), la clasificación manual final evidenció una ligera desviación: *Positive* 29.87 %, *Neutral* 36.50 % y *Negative* 33.63 %. Estas diferencias son mínimas (en promedio, menor al 3.5 % por clase), lo que indica que el modelo preentrenado tuvo un desempeño adecuado al momento de asignar etiquetas de sentimiento, logrando una distribución muy cercana a la validada por un humano.

Este resultado no solo valida parcialmente la calidad del modelo de clasificación inicial, sino que también demuestra que el dataset final conserva un nivel adecuado de balance entre clases, aspecto crítico para garantizar un entrenamiento justo y efectivo del modelo de análisis de sentimientos. Gracias a este proceso, se asegura que el modelo no se vea sesgado hacia ninguna categoría y se fortalece la representatividad del conjunto de entrenamiento.

6.1.4. Preprocesamiento de los Datos

Una vez construida la muestra que sirve como base de datos para la creación del modelo, y antes de proceder con el entrenamiento, fue necesario transformar los datos a un formato adecuado para su procesamiento automático. Este preprocesamiento incluyó tanto la conversión de las etiquetas como la preparación de los textos en una estructura numérica compatible con redes neuronales. A continuación, se describen los pasos principales:

1. **Conversión de textos y etiquetas:** Se extrajeron los textos de entrada y las etiquetas de sentimiento desde el dataset, asegurando que ambos estuvieran representados como listas de cadenas de texto.

2. **Vectorización de etiquetas:** Las etiquetas categóricas (*Negative*, *Neutral*, *Positive*) fueron transformadas a valores numéricos (0, 1 y 2 respectivamente) utilizando la clase `LabelEncoder` de `sklearn`.

3. **Tokenización del texto:** Se empleó la clase `Tokenizer` de `Keras` para construir un vocabulario y dividir cada texto en tokens (palabras individuales), asignando a cada uno un identificador numérico. Se incluyó un token especial para las palabras fuera del vocabulario (`oov_token`).

4. **Vectorización del texto:** Una vez tokenizados, los textos fueron transformados en secuencias de enteros, donde cada número representa el índice de una palabra en el vocabulario previamente construido.

5. **Normalización de longitud (Padding):** Las secuencias de tokens fueron ajustadas a una longitud fija mediante la técnica de *padding*. Este proceso rellenó o recortó las secuencias al final, garantizando que todas las entradas tuvieran la misma dimensión.

Este conjunto de transformaciones aseguró que los datos estuvieran estructurados correctamente para ser utilizados como entrada del modelo de aprendizaje profundo, permitiendo una representación eficiente y coherente del contenido textual.

6.2. Modelado y desarrollo

6.2.1. Selección

Para garantizar una selección adecuada de modelos a entrenar, se definieron criterios que equilibran aspectos técnicos, prácticos y metodológicos. Estos incluyeron: capacidad para procesar secuencias y retener información relevante, manejo del contexto, eficiencia computacional, y diversidad arquitectónica. La selección se basó en una evaluación integral considerando los recursos disponibles y las características del problema.

6.3.1.1. Criterios de Selección

Los modelos se eligieron con base en los siguientes criterios:

- **Capacidad para procesar secuencias**: El modelo debe ser capaz de manejar entradas secuenciales, como oraciones o párrafos, capturando el orden y la relación entre palabras.
- **Retención de información**: Es fundamental que el modelo retenga información relevante de pasos anteriores, especialmente en secuencias largas.
- **Eficiencia computacional**: Se priorizó un equilibrio entre desempeño y costos de entrenamiento e inferencia.
- **Diversidad arquitectónica**: Se seleccionaron modelos que representen distintos niveles de complejidad para evaluar sus ventajas relativas.

6.3.1.2. Modelos Seleccionados

Bajo estos lineamientos, se seleccionaron tres modelos recurrentes con diferentes niveles de complejidad:

- **RNN Básica:** Modelo unidireccional y simple, útil como línea base. Procesa secuencias palabra por palabra y mantiene un estado oculto actualizado en cada paso, aunque con limitaciones en la retención de información a largo plazo.
- **BiGRU:** Arquitectura bidireccional con unidades GRU que incorporan compuertas para manejar eficientemente la memoria. Captura contexto completo con menor costo computacional en comparación con LSTM.
- **BiLSTM:** Modelo más robusto, también bidireccional, con capacidad superior para retener dependencias a largo plazo y captar matices semánticos complejos, aunque con mayor carga computacional.

Todos los modelos utilizan una capa de *embedding* como entrada, seguida por la capa recurrente correspondiente y una capa de salida *softmax* para clasificación multiclase.

6.3.1.3. Resumen Comparativo de Modelos

Modelo	Complejidad	Direccionalidad	Dependencias
RNN Básica	Baja	Unidireccional	Limitado a secuencias cortas
BiGRU	Media	Bidireccional	Retención moderada gracias a compuertas
BiLSTM	Alta	Bidireccional	Alta capacidad para secuencias largas

Cuadro 6.1: Comparación cualitativa de modelos

6.2.2. Entrenamiento

6.3.2.1. Estrategia de Entrenamiento

Para desarrollar modelos adecuados a la naturaleza del problema y a las características del conjunto de datos, se adoptaron dos enfoques complementarios:

1. Variación de arquitecturas Como se menciono previamente, se exploraron tres tipos de redes neuronales recurrentes (RNN) con diferentes niveles de complejidad, con el objetivo de evaluar su capacidad de modelado contextual y su rendimiento:

- **RNN básica:** Utilizada como línea base, con bajo costo computacional.
- **GRU Bidireccional:** Equilibrio entre simplicidad y expresividad. Mejora la modelación secuencial.
- **LSTM Bidireccional:** Arquitectura más robusta, orientada a maximizar el rendimiento aunque con mayor demanda computacional.

2. Ajuste sistemático de hiperparámetros Para garantizar una comparación equitativa entre las arquitecturas, se empleó una grilla común de hiperparámetros:

- **Unidades en la capa recurrente:** 32, 64 y 128.
- **Dropout:** 0.2, 0.3 y 0.4, para regularización.
- **Batch size:** 16, 32 y 64, equilibrando estabilidad y eficiencia.
- **Epochs:** 5 y 10.

Los hiperparámetros mencionados anteriormente fueron seleccionados con el objetivo de garantizar un ajuste riguroso y alineado con las buenas prácticas descritas en la literatura especializada en aprendizaje profundo para datos secuenciales. Esta literatura recomienda considerar aspectos clave como la estabilidad durante la validación cruzada, la forma y comportamiento de las curvas de aprendizaje, el equilibrio entre precisión y *recall*, y la varianza del rendimiento entre ejecuciones [32], [33]. En este contexto, los criterios estadísticos utilizados para seleccionar las combinaciones más adecuadas de hiperparámetros fueron los siguientes:

- **Estabilidad inter-folds**: se priorizaron configuraciones con baja desviación estándar en el *F1-score macro* entre los diferentes pliegues de la validación cruzada.

- **Curvas de pérdida consistentes**: se buscó una evolución paralela entre la pérdida de entrenamiento y validación, evitando señales tempranas de sobreajuste.

- **Balance entre precisión y *recall***: se descartaron combinaciones que favorecían una métrica en detrimento de la otra.

- **Baja varianza entre ejecuciones**: se valoraron combinaciones que ofrecieran resultados estables al repetir los experimentos, minimizando la sensibilidad del rendimiento a la aleatoriedad del entrenamiento.

Estos criterios permitieron orientar el proceso de ajuste hacia configuraciones con buen rendimiento, pero también con comportamiento estable y generalizable, tal como se recomienda en prácticas estándar de modelado supervisado.

La combinación del ajuste sistematico de hiperparametros con la variación de la arquitectura, generaron configuraciones que no solo generaron un portafolio variado de modelos, sino que también facilitaron el control de problemas comunes en redes recurrentes, tales como el sobreajuste, la pérdida de capacidad de generalización y la inestabilidad en la retropropagación (desvanecimiento o explosión del gradiente).

6.3.2.2. Técnica de Entrenamiento

La técnica de entrenamiento combinó dos enfoques fundamentales: **Validación Cruzada Estratificada** y **Búsqueda Sistemática de Hiperparámetros** (*Grid Search*). Esta combinación garantizó un proceso robusto y una evaluación confiable, especialmente considerando la moderada dimensión del conjunto de datos (1011 registros). A continuación, se detallan los aspectos mas relevantes que se tuvieron en cuenta, al aplicar esta técnica.

1. Partición de datos: Se destinó el **90 %** del dataset para el conjunto de entrenamiento y validación, empleando validación cruzada estratificada. El **10 %** restante se reservó como conjunto de prueba final e independiente para medir el desempeño del modelo. Esta partición garantizó un uso eficiente de los datos y una evaluación objetiva del desempeño.

2. Validación Cruzada Estratificada: Se empleó una validación cruzada estratificada con 5 pliegues, la cual maximiza el aprovechamiento del conjunto de datos y preserva la proporción de clases en cada partición. Este enfoque proporciona estimaciones fiables del desempeño del modelo, equilibrando precisión y costo computacional de manera adecuada.

3. Búsqueda por Grilla (*Grid Search*): La búsqueda sistemática de hiperparámetros se aplicó sobre las mismas combinaciones para todas las arquitecturas evaluadas. Esto permitió comparar las configuraciones bajo condiciones homogéneas y seleccionar la que mostró mejor rendimiento.

4. Optimizador Utilizado: Para todos los entrenamientos se utilizó el optimizador Adam con la tasa de aprendizaje por defecto (0.001). Su capacidad de adaptación dinámica y rápida convergencia lo hacen especialmente adecuado para problemas de procesamiento de lenguaje natural, asegurando estabilidad y eficiencia en el aprendizaje.

6.3.2.3. Prevención de Problemas Comunes en RNN

Durante el proceso de entrenamiento se prestó especial atención a la mitigación de problemas clásicos de las redes neuronales recurrentes:

- **Sobreajuste:** Controlado mediante regularización con *dropout* y limitación del número de épocas.
- **Desvanecimiento/explosión del gradiente:** Atacado mediante la elección de arquitecturas avanzadas (GRU y LSTM) y un número razonable de unidades en la capa recurrente.
- **Pérdida de generalización:** Abordada a través de validación cruzada estratificada y reserva de un conjunto de prueba no expuesto durante el entrenamiento.

6.2.3. Evaluación

La evaluación de los modelos entrenados se realizó mediante una estrategia que integró la partición del conjunto de datos y el uso de métricas específicas para valorar su desempeño y costo computacional.

6.3.3.1. Partición de Datos para Evaluación

Como se mencionó previamente el 90 % del conjunto de datos se destinó al entrenamiento, incluyendo validación cruzada y el 10 % restante se reservó como conjunto de prueba final, garantizando una evaluación independiente y realista de la capacidad de generalización de los modelos tras el ajuste de hiperparámetros.

6.3.3.2. Métricas Utilizadas

Para medir la eficacia predictiva y la eficiencia del modelo se emplearon las siguientes métricas:

- **Accuracy:** Proporción total de predicciones correctas, ofreciendo una visión general del desempeño global.

- **Precision:** Fracción de verdaderos positivos entre todas las predicciones positivas, relevante para minimizar falsos positivos.

- **Recall:** Proporción de verdaderos positivos detectados sobre el total de positivos reales, midiendo la capacidad de detección.

- **F1-score:** Media armónica entre *Precision* y *Recall*, balanceando la precisión y la cobertura de detección.

- **Costo Computacional (horas):** Tiempo total invertido en el entrenamiento, que cuantifica el recurso computacional requerido.

Esta combinación permitió evaluar el rendimiento de los modelos desde una perspectiva integral, considerando tanto su efectividad como la viabilidad práctica de su implementación.

6.3. Resultados y análisis

6.3.1. Análisis comparativo

6.4.1.1. Métricas Globales

A continuación se presentan las métricas globales de desempeño para los modelos evaluados (RNN, BiLSTM y BiGRU):

Métrica	RNN	BiLSTM	BiGRU
Accuracy	0.8333	0.8529	0.9020
Precision	0.8357	0.8639	0.9058
Recall	0.8333	0.8560	0.9020
F1-score	0.8344	0.8572	0.9024

Cuadro 6.2: Comparación de Métricas Globales entre Modelos

Análisis: El modelo BiGRU supera a RNN y BiLSTM en todas las métricas globales. Mientras que el modelo RNN mantiene un desempeño balanceado, y el BiLSTM evidencia una mejora sobre el RNN con un *accuracy* de 0.8529 y un f1-score de 0.8572, el BiGRU destaca con un *accuracy* superior (0.9020) y una mejor integración entre precisión y sensibilidad. Esto indica que el modelo BiGRU tiene una capacidad más robusta para generalizar y clasificar correctamente.

6.4.1.2. Métricas por Clase

Clase	Modelo	Precision	Recall	F1-score
3*0 (Negativo)	RNN	0.76	0.76	0.76
	BiLSTM	0.86	0.74	0.79
	BiGRU	0.90	0.82	0.86
3*1 (Neutro)	RNN	0.76	0.78	0.77
	BiLSTM	0.76	0.86	0.81
	BiGRU	0.83	0.92	0.87
3*2 (Positivo)	RNN	1.00	0.97	0.98
	BiLSTM	0.97	0.97	0.97
	BiGRU	1.00	0.97	0.98

Cuadro 6.3: Comparación de Métricas por Clase entre Modelos

Análisis: El modelo BiGRU mejora notablemente el desempeño en las clases negativa y neutra, logrando mayores valores de precisión y recall, especialmente en la clase neutra con un recall de 0.92. El BiLSTM presenta un buen rendimiento, especialmente en la clase positiva con un f1-score de 0.97, aunque muestra un menor recall en la clase negativa y menor precisión en la clase neutra en comparación con BiGRU. Ambos modelos superan al RNN, siendo BiGRU la opción más equilibrada y precisa.

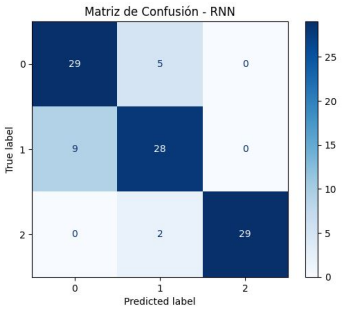


Figura 6.3: Matriz de Confusión - RNN

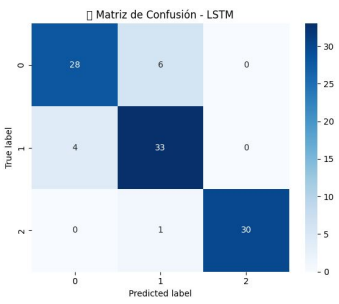


Figura 6.4: Matriz de Confusión - BiLSTM

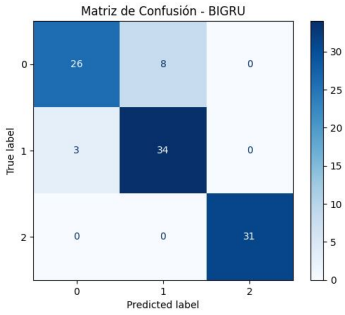


Figura 6.5: Matriz de Confusión - BiGRU

Análisis: Las matrices de confusión evidencian que BiGRU reduce significativamente las confusiones entre clases negativas y neutras presentes en el RNN y BiLSTM. Aunque BiLSTM mejora frente a RNN, todavía presenta cierta dificultad para diferenciar estas clases, mientras que el BiGRU logra una clasificación más precisa y equilibrada.

6.4.1.3. Evolución de los Modelos

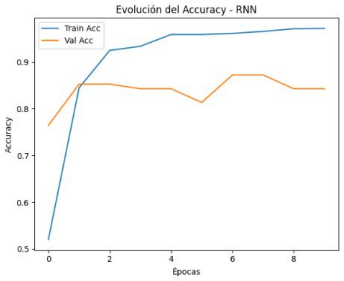


Figura 6.6: Accuracy - RNN

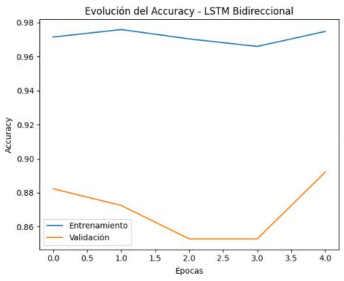


Figura 6.7: Accuracy - BiLSTM

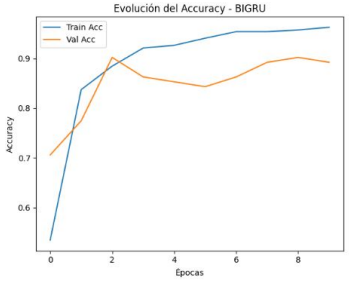


Figura 6.8: Accuracy - BiGRU

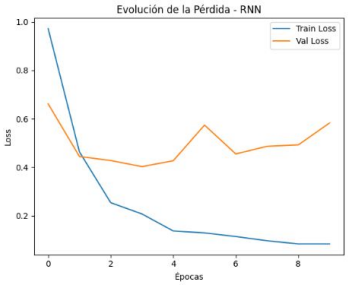


Figura 6.9: Pérdida - RNN

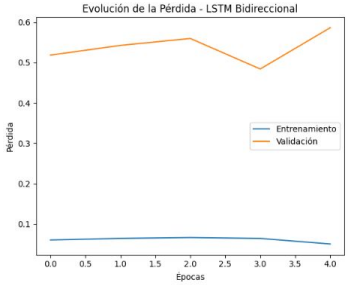


Figura 6.10: Pérdida - BiLSTM

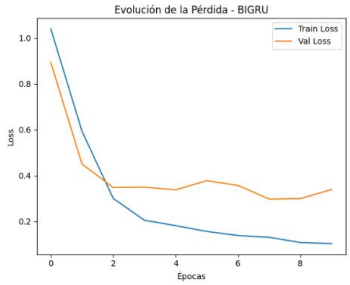


Figura 6.11: Pérdida - BiGRU

Análisis: En las tres arquitecturas, la pérdida durante el entrenamiento disminuye mostrando aprendizaje progresivo. Sin embargo, el RNN comienza a sobreajustarse a partir de la tercera época, mientras que BiLSTM y BiGRU mantienen un comportamiento más estable. El BiGRU presenta un mejor equilibrio entre desempeño en entrenamiento y validación, seguido por BiLSTM que muestra un sobreajuste moderado evidenciado por su alta precisión en entrenamiento y menor generalización. La evolución del accuracy confirma la superioridad del BiGRU y la buena performance del BiLSTM frente al RNN.

6.3.2. Modelo Seleccionado

6.4.2.1. Justificación del Modelo

El modelo seleccionado para la tarea de clasificación es el **BiGRU (Bidirectional Gated Recurrent Unit)**, debido a su superioridad en métricas clave como *accuracy*, *precision*, *recall* y *f1-score*, superando a alternativas como RNN y BiLSTM. Además, destaca por su bajo costo computacional y menor tendencia al sobreajuste, gracias a su arquitectura bidireccional que maneja eficientemente dependencias a largo plazo en datos secuenciales.

Los resultados evidencian una **accuracy** de 0.9020, **precision** de 0.9058, **recall** de 0.9020 y **f1-score** de 0.9024, junto a un tiempo de entrenamiento competitivo de 15.6 minutos, en comparación con BiLSTM de 64.2 minutos, posicionándolo como la opción más balanceada en desempeño y eficiencia computacional.

6.4.2.2. Descripción del Modelo

Como se pudo evidenciar en la sección anterior, el modelo ganador fue el **BiGRU**, que alcanzó en el conjunto de prueba un accuracy de **0.9020**, precision de **0.9058**, recall de **0.9020** y un F1-score de **0.9024**. El reporte por clase mostró un desempeño equilibrado, con un f1-score de **0.86, 0.87 y 0.98** para las clases 0, 1 y 2 respectivamente.

El modelo **BiGRU** ganador tiene dentro de su configuración **64** unidades, un dropout de **0.2**, batch size de **16** y fue entrenado durante **10** épocas. Esta configuración mostró una excelente capacidad de generalización en un tiempo de entrenamiento total de apenas **15.6** minutos, lo que refleja su eficiencia computacional sin comprometer el rendimiento.

A partir de esta base sólida, se llevaron a cabo ajustes adicionales con el objetivo de **mejorar aún más el rendimiento y la robustez del modelo**. A continuación, se describen las principales mejoras realizadas.

6.4.2.3. Mejoras al Modelo

6.4.2.4.1. Descripción del proceso de mejora: Tras observar un desempeño competitivo pero con indicios de sobreajuste en el modelo **BiGRU**, se propuso una serie de mejoras con los siguientes objetivos:

- 1. Incrementar la capacidad de generalización del modelo y reducir el sobreajuste.
- 2. Explorar un rango más amplio de configuraciones mediante la expansión de la grilla de hiperparámetros.
- 3. Mejorar las métricas de evaluación mediante ajustes finos y técnicas de regularización.

Para alcanzar estos objetivos, se implementaron las siguientes acciones:

- **Expansión de la grilla de hiperparámetros**, incluyendo:

Parámetro	Antes (Grilla Original)	Después (Grilla Ampliada)
`units`	[64, 128]	[32, 64, 128]
`dropout`	[0.2, 0.3]	[0.2, 0.3, 0.4]
`batch_size`	[32, 64]	[16, 32, 64]
`epochs`	[10, 20]	[5, 10, 20, 50]
`learning_rate`	No considerado	[0.001, 0.0005]

Cuadro 6.4: Comparación de grilla de hiperparámetros antes y después de la mejora

- **Incorporación del parámetro** `learning_rate`, que permitió un mejor ajuste del optimizador y mayor estabilidad en el entrenamiento.
- **Implementación de *EarlyStopping*** con monitoreo sobre `val_loss` y paciencia de 5 épocas, lo que permitió detener el entrenamiento en el punto óptimo y restaurar los mejores pesos del modelo.

6.4.2.4.2. Resultados comparativos: BiGRU Original vs Mejorado La siguiente tabla resume las métricas de evaluación globales y por clase, comparando el modelo inicial con la versión mejorada:

Métrica	Inicial	Mejorado	Mejora (%)
Accuracy	90.20 %	92.16 %	+2.18 %
Precisión	90.58 %	92.70 %	+2.42 %
Recall	90.20 %	92.30 %	+2.43 %
F1-Score	90.24 %	92.46 %	+2.44 %
F1 Clase 0	86.00 %	90.00 %	+4.65 %
F1 Clase 1	87.00 %	89.00 %	+2.30 %
F1 Clase 2	98.00 %	98.00 %	0.00 %
Tiempo Entreno (min)	15.6	44.4	+184.62 %

Cuadro 6.5: Comparación de desempeño: Modelo Inicial vs. Modelo Mejorado

Parámetro	Modelo Seleccionado	Modelo Mejorado
Unidades (Neuronas)	64	32
Dropout	0.2	0.3
Batch size	16	16
Epochs	10	20
Learning rate	–	0.001

Cuadro 6.6: Comparación de configuración óptima entre modelos

Beneficios de la mejora:

- Gracias al uso de **EarlyStopping**, se logró mitigar de manera efectiva el sobreajuste presente en las primeras versiones del modelo, favoreciendo una mejor generalización.

- Todas las métricas globales experimentaron incrementos significativos: el **accuracy** mejoró en un 2.18 %, la **precisión** aumentó un 2.42 %, el **recall** subió un 2.43 % y el **F1-score** global creció un 2.44 %, reflejando un avance real y consistente en el desempeño.

- En las clases específicas, se destaca la mejora del F1-score de la clase 0 en un 4.65 %, y de la clase 1 en un 2.30 %, contribuyendo a una clasificación más equilibrada y precisa.

- La clase positiva (clase 2), crítica en muchas aplicaciones, mantuvo un rendimiento excelente con un F1-score perfecto de 0.98, asegurando alta confiabilidad en predicciones relevantes.

- Aunque el tiempo de entrenamiento aumentó un 184.62 %, esta inversión se traduce en un modelo mucho más robusto y preciso, lo cual es fundamental para aplicaciones de alta exigencia.

Conclusión: La optimización del modelo BiGRU mediante una grilla de hiperparámetros más amplia y la implementación de *EarlyStopping* resultó en mejoras significativas en precisión, cobertura y equilibrio general. Estas mejoras posicionan al modelo mejorado como una solución más robusta, confiable y generalizable en comparación con su versión original. En vista de estos resultados positivos, el modelo BiGRU mejorado fue seleccionado para la clasificación de todos los registros de la base de datos, asegurando así un análisis más preciso y consistente en las etapas posteriores del proyecto.

6.3.3. Conclusiones

- **Modelo seleccionado:** El modelo BiGRU demostró un desempeño superior frente a BiLSTM y RNN en métricas clave como accuracy, precision, recall y f1-score, consolidándose como la mejor opción para la clasificación de sentimientos en este estudio.

- **Balance entre desempeño y eficiencia:** BiGRU combina alta precisión con un costo computacional competitivo y menor riesgo de sobreajuste, gracias a su arquitectura bidireccional que optimiza el manejo de dependencias secuenciales a largo plazo.

- **Impacto de la optimización de hiperparámetros:** La ampliación de la búsqueda de hiperparámetros y la incorporación de técnicas como EarlyStopping permitieron mejorar significativamente el rendimiento del BiGRU, alcanzando un f1-score superior a 0.92 y una mayor estabilidad en el entrenamiento.

- **Capacidad de generalización y robustez:** El análisis por clase evidenció un desempeño equilibrado en todas las categorías, especialmente en la clase positiva, con una alta capacidad de generalización y menor tendencia al sobreajuste en el modelo final.
- **Recomendación práctica:** Basado en los resultados, se recomienda implementar el modelo BiGRU para soluciones de clasificación de sentimientos robustas y eficientes, considerando a BiLSTM como una alternativa válida cuando se requiera un modelo con buen equilibrio y capacidad contextual.

Capítulo 7

Adaptación de Modelos LLM para Interacción con el Cliente

Con el propósito de capacitar al sistema para realizar un análisis inteligente y automatizado de las interacciones con los clientes, este capítulo explora el proceso de *fine-tuning* aplicado a modelos de lenguaje de gran escala (LLMs) especializados en el ámbito de la atención al cliente. El objetivo es dotar al modelo no solo de la habilidad para identificar emociones, tendencias y puntos críticos en las conversaciones, sino también para responder con precisión y empatía a cada solicitud. Para ello, se emplean técnicas de *fine-tuning* sobre modelos preentrenados, utilizando datos específicos del contexto CRM, apoyándose en la base de datos mencionada previamente para este fin.

7.1. Limpieza y Preprocesamiento

7.1.1. Descripción del Dataset Inicial

El dataset utilizado en esta etapa corresponde al mismo conjunto de datos descrito previamente en el Capítulo 5, el cual contiene interacciones textuales entre usuarios y un sistema de atención al cliente. En esta fase, dicho dataset fue ampliado mediante la incorporación de una nueva variable: el **sentimiento asociado a cada instrucción del usuario**, el cual fue inferido utilizando el modelo de análisis de sentimientos desarrollado con una muestra previamente etiquetada de 1.011 registros.

De esta manera, el nuevo dataset quedó conformado por tres variables: (1) **Instrucción del usuario**, (2) **Sentimiento de la instrucción**, y (3) **Respuesta del sistema**. Esta versión enriquecida del dataset fue la base para las tareas posteriores de entrenamiento y ajuste fino de modelos de lenguaje.

7.1.2. Limpieza de Datos

Con el objetivo de simplificar y estandarizar los textos, mejorar su calidad semántica y asegurar su adecuación para el proceso de entrenamiento mediante fine-tuning, se llevó a cabo una fase de limpieza del conjunto de datos. A continuación, se detallan las principales acciones implementadas:

- **Eliminación de registros duplicados**: Durante el análisis exploratorio del conjunto de datos, se identificaron **8,168 registros duplicados** en la variable correspondiente a las entradas del usuario. Esta redundancia representaba un riesgo importante de sobreajuste y sesgo durante el entrenamiento del modelo. Para abordar este problema, se implementó una técnica de *deduplicación semántica basada en embeddings*, que permitió reducir el conjunto original de **26,872 muestras** a **18,704 ejemplos únicos**.

A diferencia de los métodos tradicionales que eliminan duplicados mediante coincidencia exacta de texto, esta técnica se fundamenta en la comparación de *similitud semántica* utilizando modelos de embeddings de oraciones. Específicamente, se empleó el modelo `SentenceTransformer` propuesto por Reimers y Gurevych (2019), el cual convierte cada oración en un vector denso dentro de un espacio semántico, permitiendo comparar frases similares en significado aunque difieran en estructura textual.

El procedimiento aplicado fue el siguiente:

1. **Cálculo de embeddings**: Se generaron vectores semánticos para todas las respuestas del conjunto de entrenamiento utilizando un modelo preentrenado de SentenceTransformers (e.g., `all-MiniLM-L6-v2`).

2. **Normalización de vectores**: Los embeddings fueron normalizados para permitir el cálculo eficiente de la similitud por coseno.

3. **Indexación y búsqueda**: Se empleó la librería `FAISS` (Johnson et al., 2017) para indexar los vectores y realizar búsquedas de vecinos más cercanos de forma eficiente.

4. **Filtro por umbral de similitud**: Para cada oración, se buscó su vecino más cercano y se eliminó uno de los dos si la similitud superaba el umbral de 0,95.

5. **Selección final**: Se construyó un nuevo conjunto de datos manteniendo únicamente los ejemplos únicos desde el punto de vista semántico.

Esta estrategia mejoró la representatividad del conjunto de entrenamiento, asegurando que cada muestra aportara información distinta, lo que redujo la redundancia y fortaleció la capacidad generalizadora del modelo.

7.1.3. Construcción del Dataset para el Modelado

Una vez finalizado el proceso de limpieza y depuración, se procedió a estructurar el conjunto de datos para su uso en el entrenamiento del modelo de lenguaje.

1. División del Conjunto de Datos: El conjunto completo fue aleatorizado para evitar sesgos por ordenamiento previo, y posteriormente dividido en tres subconjuntos de acuerdo con las mejores prácticas de entrenamiento:

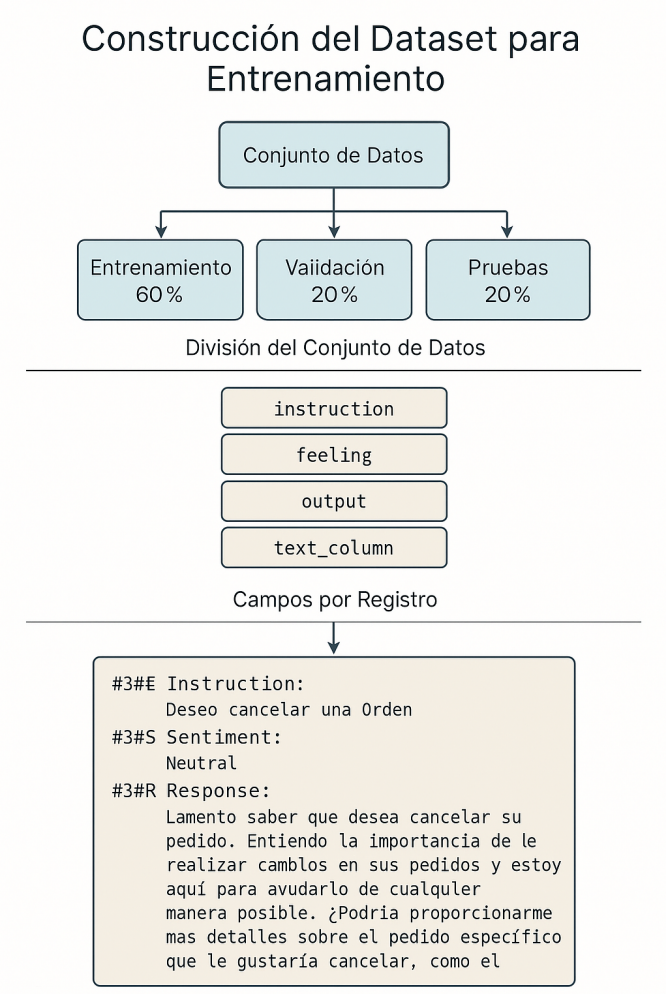


Figura 7.1: Diagrama del proceso de construcción del dataset

- **Entrenamiento (60 %):** Datos utilizados directamente para el aprendizaje del modelo.
- **Validación (20 %):** Datos empleados durante el entrenamiento para monitorear el rendimiento y evitar sobreajuste.
- **Prueba (20 %):** Datos no vistos por el modelo, usados para evaluar su capacidad de generalización al finalizar el proceso de entrenamiento.

2. Definición de Estructura por Registro: Cada registro del conjunto de datos fue estructurado con los siguientes campos clave:

- `instruction`: Entrada textual original del usuario (por ejemplo, una solicitud o pregunta).
- `feeling`: Sentimiento asociado a la instrucción, obtenido mediante una etapa previa de análisis de emociones.
- `output`: Respuesta esperada correspondiente a la instrucción y al sentimiento detectado.
- `text_column`: Campo auxiliar que conserva el texto completo del registro original.

3. Formateo para Entrenamiento: El siguiente paso consistió en transformar estos campos en un formato consolidado que simula una conversación estructurada. Este formato fue diseñado para alinear la entrada del modelo con la tarea de generación condicional. Para ello, se utilizó la función `chat_template`, la cual genera un texto con la siguiente plantilla:

```
### Instruction:
Deseo cancelar una Orden

### Sentiment:
Neutral

### Response:
Lamento saber que desea cancelar su pedido.
```

Este esquema no solo facilita el entrenamiento de modelos autoregresivos, sino que también orienta al modelo sobre la estructura esperada de la interacción, integrando la instrucción, el estado emocional y la respuesta.

En conjunto, esta fase permitió transformar un corpus textual en bruto en un conjunto de datos formalizado y optimizado para el entrenamiento de modelos de lenguaje adaptados al contexto de atención al cliente.

7.2. Modelado y Desarrollo

7.2.1. Selección

Para el desarrollo del sistema de procesamiento de lenguaje natural (PLN) enfocado en la atención automatizada al cliente, se realizó un proceso de evaluación riguroso basado en criterios técnicos, operativos y contextuales. Este proceso tuvo como objetivo identificar un modelo de lenguaje de gran escala (LLM) de acceso

abierto que se ajustara de manera óptima a los objetivos del proyecto, permitiendo tanto su adaptación mediante técnicas de ajuste fino como su despliegue en entornos con limitaciones de infraestructura.

7.2.1.1. Criterios de Selección

Se definieron los siguientes criterios para orientar la elección del modelo, garantizando su viabilidad técnica y alineación con el contexto de aplicación:

- **Acceso abierto:** Se priorizaron modelos de código abierto que pudieran ser replicables, auditables y desplegables sin restricciones comerciales ni dependencia de licencias propietarias.
- **Adaptabilidad y eficiencia operativa:** Capacidad del modelo para ejecutarse en infraestructuras heterogéneas, incluyendo entornos con recursos computacionales limitados, especialmente relevantes en contextos latinoamericanos.
- **Arquitectura moderna y comunidad activa:** Se valoró la existencia de documentación extensa, mantenimiento activo y una comunidad técnica que respalde el uso, entrenamiento y ajuste del modelo.
- **Compatibilidad con fine-tuning eficiente:** Preferencia por modelos que admitan técnicas de ajuste fino optimizadas, como LoRA y QLoRA, que permiten una personalización eficiente con menor costo computacional.
- **Desempeño en tareas de PLN:** Evidencia de rendimiento competitivo en tareas clave de procesamiento de lenguaje natural, como clasificación, generación y comprensión contextual.

7.2.1.2. Modelos Seleccionados

Con base en los criterios anteriores, se analizaron tres modelos LLM de acceso abierto: **LLaMA2**, **PHI** y **Gemma**. A continuación, se describen sus principales características y los motivos que influyeron en su selección o descarte:

- **LLaMA2 (Meta):** Fue el modelo seleccionado como base del sistema por su excelente equilibrio entre rendimiento, eficiencia y apertura. Su arquitectura moderna está optimizada para entornos de computación distribuida y es ampliamente compatible con técnicas de fine-tuning como LoRA y QLoRA. Además, su licencia abierta y documentación robusta lo hacen ideal para desarrollos replicables y sostenibles [34].
- **PHI (Microsoft):** Aunque presenta un enfoque interesante hacia la personalización de respuestas con base en el contexto emocional del usuario, su disponibilidad está limitada a versiones ligeras y carece de documentación extensa. Estas limitaciones dificultaron su integración plena en el marco del presente proyecto [35].

- **Gemma (Google):** Este modelo muestra una arquitectura optimizada para generación textual con coherencia semántica. Sin embargo, su documentación es escasa y su rendimiento fue inferior en las pruebas preliminares, por lo que fue descartado como modelo principal.

7.2.1.3. Resumen Comparativo de Modelos

A continuación, se presenta un resumen de los modelos evaluados, destacando sus características más relevantes y la justificación para su elección o descarte en este proyecto:

Modelo	Empresa desarrolladora	Características y evaluación
LLaMA2	Meta	Modelo seleccionado. Código abierto, rendimiento competitivo en comprensión y generación, alta compatibilidad con técnicas modernas de fine-tuning (LoRA, QLoRA), eficiente en recursos y con amplia comunidad de soporte.
PHI	Microsoft	Orientado a interacción personalizada. Limitado acceso a versiones completas y escasa documentación dificultaron su adopción en el presente proyecto.
Gemma	Google	Arquitectura moderna orientada a generación semántica. Sin embargo, presentó menor rendimiento y documentación limitada.

Cuadro 7.1: Resumen comparativo de los modelos LLM evaluados

Con base en este análisis, **LLaMA2 fue seleccionado como modelo principal para el desarrollo del sistema**, debido a su combinación de rendimiento técnico, apertura y facilidad de personalización, lo que lo convierte en una alternativa robusta, escalable y alineada con los objetivos del proyecto.

7.2.2. Entrenamiento

El ajuste fino (fine-tuning) del modelo constituye una etapa crítica en el desarrollo de sistemas de procesamiento de lenguaje natural (PLN), ya que define en gran medida la capacidad del modelo para adaptarse a contextos específicos. En este proyecto, se implementó un enfoque estratégico centrado en la eficiencia computacional y la escalabilidad, lo cual permitió realizar el entrenamiento incluso con recursos de hardware limitados.

7.2.2.1. Estrategia de Entrenamiento

La estrategia adoptada combinó técnicas modernas de ajuste fino, diseñadas específicamente para reducir el consumo de memoria y acelerar los tiempos de

entrenamiento sin comprometer la calidad del modelo. Se optó por emplear LoRA (Low-Rank Adaptation) en conjunto con QLoRA, una variante optimizada para modelos cuantizados, permitiendo trabajar con modelos de gran tamaño como LLaMA2-7B en una única GPU con 24 GB de VRAM.

7.2.2.2. Técnica de Entrenamiento

1. Adaptación mediante LoRA LoRA consiste en insertar matrices de bajo rango en capas específicas del modelo, manteniendo congelados los parámetros originales. Esta técnica reduce significativamente la cantidad de parámetros entrenables, lo que conlleva:

- Disminución del uso de memoria.
- Mayor velocidad de entrenamiento.
- Estabilidad en el ajuste y posibilidad de reutilizar el modelo base.

Los parámetros utilizados incluyeron un rango (`r`) de 8, configuración para tareas de lenguaje autoregresivo (`task_type = CAUSAL_LM`), y la selección de módulos como `q_proj`, `k_proj` y `v_proj` como objetivos del ajuste.

2. Optimización con QLoRA QLoRA permite entrenar modelos en formato cuantizado de 4 bits, lo que reduce significativamente el uso de memoria. Se empleó la clase `BitsAndBytesConfig` con los siguientes parámetros clave:

- `load_in_4bit=True`: carga en baja precisión.
- `bnb_4bit_quant_type="nf4"`: tipo de cuantización.
- `bnb_4bit_compute_dtype=torch.float16`: precisión en los cálculos.
- `bnb_4bit_use_double_quant=True`: doble cuantización para menor uso de memoria.

La carga del modelo se configuró con asignación automática de dispositivos y técnicas adicionales como `gradient_checkpointing` para reducir el uso de memoria durante el retropropagado.

3. Hiperparámetros Para determinar la configuración óptima de hiperparámetros, se realizó una **parrilla de búsqueda** (*grid search*) que evaluó distintas combinaciones de valores relevantes. Las pruebas se centraron en lograr una convergencia estable, minimizar la pérdida de entrenamiento y evitar el sobreajuste.

Ép.	BT	BE	LR	WD	Clip	Warmup	Sched.
3	2	4	1×10^{-4}	0.01	0.3	3 %	cosine_with_restarts
3	4	4	5×10^{-5}	0.01	1.0	5 %	linear
4	2	2	1×10^{-5}	0.001	0.3	10 %	constant
2	2	4	2×10^{-4}	0.01	0.5	0 %	cosine

Cuadro 7.2: Parrilla de búsqueda de hiperparámetros evaluada

La configuración seleccionada (primera fila) ofreció el mejor rendimiento dentro del entorno descrito. Mostró una convergencia rápida, bajo consumo de memoria y resultados estables, lo que la hizo ideal para su implementación final en el proceso de entrenamiento de `LLaMA2-7B`.

4.1. Descripción de los hiperparámetros utilizados Cada hiperparámetro cumple una función específica dentro del proceso de entrenamiento. A continuación, se detallan sus propósitos:

- **Épocas de entrenamiento:** Número de veces que el modelo recorre todo el conjunto de entrenamiento. Un número bajo puede llevar a un modelo subentrenado, mientras que uno muy alto puede provocar sobreajuste.

- **Tamaño de batch de entrenamiento y evaluación:** Define cuántos ejemplos se procesan simultáneamente en cada iteración. Un tamaño pequeño reduce el uso de memoria, pero puede hacer que el entrenamiento sea más ruidoso y lento. En cambio, un tamaño mayor mejora la estabilidad, pero requiere más memoria.

- **Tasa de aprendizaje (*learning rate*):** Controla qué tan grandes son los pasos que da el modelo al ajustar sus parámetros. Una tasa demasiado alta puede hacer que el modelo no converja; una muy baja puede hacer que el entrenamiento sea muy lento o quede atrapado en mínimos locales.

- **Decaimiento del peso (*weight decay*):** Técnica de regularización que penaliza grandes valores en los pesos del modelo, ayudando a evitar el sobreajuste y mejorando la generalización.

- **Recorte de gradientes (*gradient clipping*):** Establece un límite máximo al valor de los gradientes para evitar explosiones durante la retropropagación, especialmente útil en modelos grandes o con batches pequeños.

- **Calentamiento de la tasa de aprendizaje (*warmup*):** Durante las primeras iteraciones, se incrementa progresivamente la tasa de aprendizaje para estabilizar el entrenamiento y evitar grandes actualizaciones iniciales que puedan dañar el modelo.

- **Planificador de tasa de aprendizaje (*learning rate scheduler*):** Define cómo cambia la tasa de aprendizaje a lo largo del entrenamiento. En este caso, se usó `cosine_with_restarts`, que reduce gradualmente la tasa con oscilaciones periódicas, ayudando a salir de mínimos locales y a mejorar la exploración.

La combinación de estos hiperparámetros fue cuidadosamente ajustada para maximizar el rendimiento y la estabilidad del modelo, teniendo en cuenta tanto el comportamiento observado durante el entrenamiento como las limitaciones del entorno computacional disponible.
Este resultado destaca la importancia de adaptar los hiperparámetros no solo al modelo, sino también a las capacidades del sistema de entrenamiento, maximizando así el uso eficiente del hardware disponible.

4. Optimizador Utilizado Se empleó el optimizador **Adam** por su capacidad de adaptación dinámica y robustez en escenarios con ruido o gradientes dispersos, lo que favorece una convergencia más rápida y estable.

5. Recursos Utilizado El proceso de entrenamiento del modelo se llevó a cabo en un entorno local de alto rendimiento configurado específicamente para el ajuste fino de modelos de lenguaje de gran escala. Las características técnicas del sistema fueron las siguientes:

- **Versión de PyTorch:** 2.5.1 + cu124
- **Versión de PyTorch Lightning:** 2.5.0.post0
- **Versión de CUDA:** 12.6 (nvcc)
- **Versión de cuDNN:** 9.0.1
- **GPU disponible:** NVIDIA GeForce RTX 3090 Ti (24 GB VRAM)

Este entorno fue suficientemente robusto para permitir el entrenamiento de modelos como `LLaMA2-7B` utilizando técnicas de optimización de recursos como LoRA y QLoRA.

7.2.2.3. Prevención de Problemas Comunes durante el entrenamiento

Durante el entrenamiento, se aplicaron medidas para evitar problemas típicos de ajuste fino en modelos grandes, como errores de `Out of Memory`. Se utilizaron técnicas de reducción de memoria como `gradient_checkpointing` y el desactivado de caché con `use_cache=False`, lo que permitió mantener la estabilidad del proceso.

7.3. Evaluación

7.3.1. Criterios de evaluación

Para seleccionar el modelo más adecuado para el sistema de procesamiento de lenguaje natural, se definieron criterios que garantizan tanto la calidad de la generación de texto como la coherencia y naturalidad de las respuestas. Estos criterios son esenciales para asegurar que el modelo entregue resultados precisos y útiles en el contexto de atención al cliente, donde la interacción debe ser fluida y empática. Se priorizó la capacidad del modelo para producir texto que se asemeje al lenguaje humano y mantener la fidelidad al mensaje original, además de la eficiencia en la generación de respuestas con baja incertidumbre.

Los criterios principales fueron:

- **Precisión en la generación de texto:** La habilidad del modelo para reproducir de manera exacta el contenido esperado, medido por métricas que comparan con textos de referencia.
- **Coherencia y fidelidad semántica:** La capacidad de mantener el sentido y la estructura del mensaje original, fundamental para respuestas completas y relevantes.
- **Fluidez y naturalidad:** Refleja qué tan natural y comprensible es el texto generado, lo que impacta directamente en la experiencia del usuario final.

Estos criterios guiaron la selección de las métricas de evaluación empleadas para medir cuantitativamente el desempeño de los modelos evaluados.

7.3.2. Métricas de evaluación

Para cuantificar el desempeño de los tres modelos evaluados, se utilizaron las métricas clásicas ampliamente reconocidas en el procesamiento de lenguaje natural:

1. **BLEU Score**
   El *BLEU (Bilingual Evaluation Understudy) Score* es una métrica estándar para evaluar la calidad del texto generado, especialmente en traducción automática [36]. Se basa en la comparación de n-gramas entre la salida del modelo y una o varias referencias humanas. El puntaje oscila entre 0 y 1, donde valores cercanos a 1 indican una alta coincidencia con el texto de referencia, reflejando precisión y exactitud en la generación.

2. **ROUGE-L Score**
   El *ROUGE (Recall-Oriented Understudy for Gisting Evaluation) Score*, particularmente la variante *ROUGE-L*, evalúa la calidad del texto generado mediante la subsecuencia común más larga (Longest Common Subsequence, LCS) con respecto a una referencia [37]. Al igual que BLEU, varía entre 0 y 1, donde valores cercanos a 1 indican que el modelo genera respuestas coherentes y fieles al contenido original, manteniendo la integridad semántica.

3. **Perplejidad (PPL)**
   La *Perplejidad* mide la incertidumbre del modelo al predecir la siguiente palabra en una secuencia [38]. Un valor mínimo de 1 indica predicción perfecta y aumenta con la incertidumbre. Por tanto, un valor bajo de perplejidad sugiere que el modelo genera texto más coherente, natural y fluido, mejorando la experiencia del usuario.

Para complementar esta evaluación tradicional, se utilizó además la métrica *RAGAS* (Recall and Generative Assessment Score) — una medida más reciente que combina aspectos de recuperación y generación para evaluar la capacidad del modelo de producir respuestas relevantes y bien fundamentadas. Dado el alto costo computacional asociado a esta métrica, su uso se limitó únicamente al modelo seleccionado para despliegue, **LLaMA2 - 7B**.

7.4. Resultados y análisis

7.4.1. Evaluación con metricas clásicas

Los modelos *LLama2-7B*, *Gemma-2B* y *Phi-8B* fueron evaluados utilizando las métricas clásicas BLEU, ROUGE-L y Perplejidad (PPL), cuyos resultados se resumen en la Tabla 7.3.

Modelo	**BLEU Score**	**ROUGE-L**	**Perplejidad (PPL)**
LLama2 - 7B	12.25	0.34	7.3
Gemma - 2B	10.33	0.13	15.6
Phi - 8B	8.56	0.45	16.8

Cuadro 7.3: Resultados de la evaluación de los modelos utilizando las métricas BLEU, ROUGE-L y Perplejidad.

El análisis indica que *LLama2 - 7B* presenta la mayor precisión en generación, mientras que *Phi - 8B* sobresale en coherencia semántica. En cuanto a fluidez, medida a través de la perplejidad, *LLama2 - 7B* también obtiene el mejor resultado, sugiriendo respuestas más naturales.

7.4.2. Evaluación complementaria con métricas modernas

Para complementar la evaluación, se utilizó el conjunto de métricas provisto por **RAGAS** (Retrieval-Augmented Generation Assessment Score) [25] . Este framework permite analizar modelos de generación condicional desde múltiples perspectivas, combinando métricas clásicas de PLN con criterios modernos centrados en fidelidad, relevancia y precisión contextual.

La Figura 7.2 muestra los puntajes obtenidos en una muestra representativa de 500 ejemplos generados por el modelo `LLaMA2-7B` ajustado con LoRA + QLoRA.

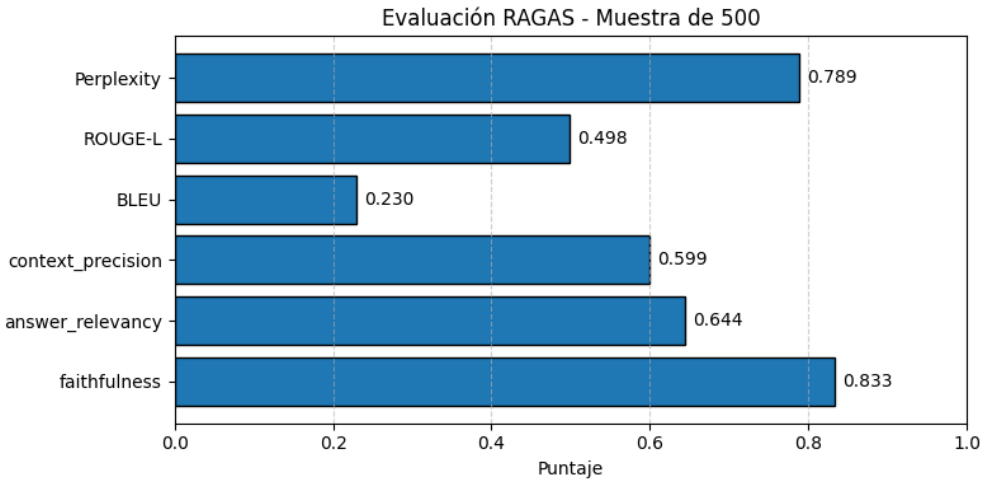


Figura 7.2: Evaluación del modelo mediante métricas RAGAS (muestra de 500)

Análisis de resultados:

- **Faithfulness (0.833):** La métrica más destacada. Indica que el modelo mantiene una alta coherencia entre la información generada y el contexto disponible. Este valor es clave en aplicaciones donde es importante no inventar datos o tergiversar información.

- **Answer Relevancy (0.644):** El modelo tiende a generar respuestas relevantes a la intención del usuario. Aunque no es perfecto, este resultado sugiere un buen alineamiento entre entrada y salida, lo que es vital en escenarios de atención al cliente.

- **Context Precision (0.599):** Refleja la capacidad del modelo para aprovechar adecuadamente el contexto en la generación de respuestas. Si bien el resultado es moderado, se mantiene dentro de rangos aceptables, especialmente considerando el ajuste con recursos limitados.

- **Perplexity (0.789):** Esta métrica, tradicional en modelos de lenguaje, indica una buena capacidad del modelo para generar secuencias lingüísticamente plausibles.

- **ROUGE-L (0.498):** Mide el solapamiento de unidades de texto entre la respuesta generada y la referencia. El puntaje es aceptable, aunque evidencia espacio para mejoras en cobertura léxica.

- **BLEU (0.230):** Métrica más baja del conjunto, refleja la dificultad de igualar exactamente la redacción esperada. Esta limitación es común en tareas generativas con alta variabilidad expresiva.

7.4.3. Monitoreo del ajuste fino mediante QLoRA

7.4.3.1. Evolución de la Pérdida durante el Entrenamiento

Durante el ajuste fino del modelo *LLaMA2 - 7B* con la técnica QLoRA, se registró la evolución de la pérdida en entrenamiento y validación (Tabla 7.4), con el fin de evaluar el aprendizaje y detectar posibles problemas como sobreajuste.

Step	Training Loss	Validation Loss
100	1.9992	1.8826
200	1.5786	1.5010
300	1.3417	1.3140
400	1.2470	1.2257
500	1.1307	1.1111
600	1.0660	1.0393
700	0.9972	0.9947
800	0.9823	0.9752
900	0.9771	0.9609
1000	0.9614	0.9519
1100	0.9607	0.9460
1200	0.9451	0.9429
1300	0.9424	0.9413
1400	0.9399	0.9409

Cuadro 7.4: Evolución de la pérdida durante el entrenamiento con QLoRA

La Figura 7.3 presenta la evolución de la función de pérdida (*loss*) durante el entrenamiento del modelo `LLaMA2-7B` utilizando la técnica **QLoRA**. Se muestran tanto la pérdida sobre el conjunto de entrenamiento (*Training Loss*) como sobre el conjunto de validación (*Validation Loss*), en función del número de pasos de entrenamiento (*steps*).

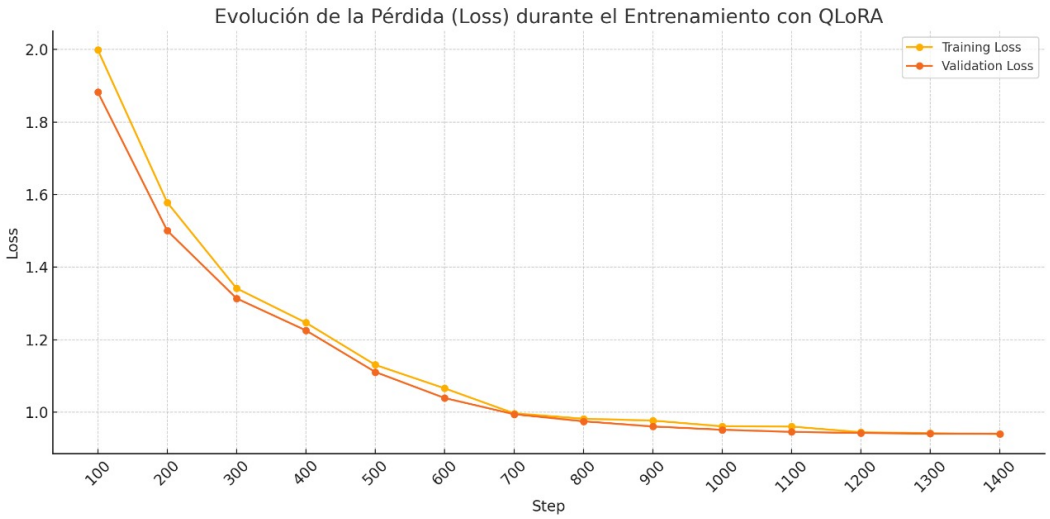


Figura 7.3: Evolución de la pérdida durante el entrenamiento con QLoRA

Análisis de la gráfica:

- En las primeras etapas del entrenamiento (hasta aproximadamente el paso 300), se observa una disminución acelerada de la pérdida, lo cual indica que el modelo está aprendiendo patrones relevantes desde los primeros ejemplos.
- A partir del paso 300, la pérdida comienza a descender de forma más progresiva, señalando una fase de refinamiento de los parámetros internos. Este comportamiento es esperado en modelos bien entrenados y sugiere una adecuada tasa de aprendizaje.
- La cercanía entre las curvas de **training** y **validation** a lo largo de todo el proceso refleja una buena generalización, sin presencia de sobreajuste significativo (*overfitting*), incluso hacia el final del entrenamiento.
- Hacia los últimos pasos (aproximadamente 1200 a 1400), la función de pérdida tiende a estabilizarse en torno a valores cercanos a 0.95–1.0, lo cual indica una convergencia adecuada del modelo.

Estos resultados confirman que los hiperparámetros seleccionados (ver Sección 7.2) y la combinación de técnicas de ajuste (LoRA + QLoRA) permitieron un entrenamiento estable y eficiente, garantizando un equilibrio entre rendimiento y uso de recursos computacionales.

7.4.3.3. Ventajas Observadas

La combinación de LoRA y QLoRA permitió:

- Reducción de más del 70 % del uso de memoria.
- Conservación de calidad en comparación con modelos sin cuantización.

- Entrenamiento exitoso de LLaMA2-7B en una sola GPU de 24 GB.

El proceso de entrenamiento fue exitoso gracias a una estrategia técnica que equilibró rendimiento, eficiencia y viabilidad operativa. Esto permitió implementar un modelo grande y potente con recursos accesibles, sin sacrificar la calidad del ajuste ni la posibilidad de replicación.

7.4.4. Evaluación Comparativa de la Generación de Respuestas con y sin Sentimiento

Con el fin de analizar el impacto de la variable **sentimiento** en la calidad de las respuestas generadas por el modelo de lenguaje seleccionado (LLaMA2-7B), se llevó a cabo una evaluación comparativa del modelo bajo dos configuraciones distintas:

- **Escenario 1: Con sentimiento**. El modelo fue entrenado utilizando como entrada tanto la instrucción del usuario como el sentimiento inferido por el modelo de análisis de sentimiento creado (positivo, negativo o neutro). Esta configuración ya fue evaluada previamente mediante las métricas BLEU, ROUGE-L, Perplexity y RAGAS, cuyos resultados se presentaron en la sección anterior.
- **Escenario 2: Sin sentimiento**. Se utilizó el mismo modelo base, pero en este caso solo se consideró la instrucción del usuario como entrada, omitiendo cualquier información relacionada con el sentimiento.

Ambos escenarios fueron evaluados utilizando el mismo conjunto de datos y las mismas métricas previamente definidas. Estas incluyen indicadores de coincidencia superficial como **BLEU**, **ROUGE-L** y **Perplexity**, así como tres métricas semánticas del conjunto **RAGAS** (*Retrieval-Augmented Generation Assessment Score*): *Faithfulness*, *Answer Relevancy* y *Context Precision*. A continuación, se presentan los resultados de dicha comparación.

7.4.4.1. Resultados comparativos

La siguiente tabla resume los valores obtenidos para ambas configuraciones, así como la variación porcentual relativa entre el modelo con y sin sentimiento:

Métrica	Sin Sentimiento	Con Sentimiento	Variación (%)
BLEU	0,310	0,230	-25,81 %
ROUGE-L	0,528	0,498	-5,68 %
Perplexity	0,775	0,789	+1,81 %
Faithfulness	0,749	**0,833**	**+11,21 %**
Answer Relevancy	0,631	**0,644**	**+2,06 %**
Context Precision	0,465	**0,599**	**+28,82 %**

Cuadro 7.5: Comparación de métricas entre modelos con y sin variable de sentimiento

7.4.4.2. Análisis y conclusión

Los resultados obtenidos evidencian un impacto significativo al incorporar la variable sentimiento en el proceso de generación de respuestas. Si bien, las métricas de coincidencia superficial como **BLEU** y **ROUGE-L** presentan una disminución del 25,81 % y 5,68 % respectivamente, y la **Perplexity** se incrementa en un 1,81 % (lo que sugiere una ligera pérdida de fluidez), estas variaciones resultan menos críticas dado el propósito del proyecto, que se enfoca en desarrollar un sistema de atención conversacional más empático, preciso y adaptado al contexto del usuario, las métricas semánticas y contextuales del conjunto **RAGAS** adquieren mayor relevancia. En este sentido, el modelo con sentimiento mostró mejoras sustanciales:

- **Faithfulness** aumentó un 11,21 %, indicando mayor fidelidad a la evidencia proporcionada.
- **Answer Relevancy** mejoró un 2,06 %, reflejando una alineación más precisa con la intención del usuario.
- **Context Precision** incrementó en un 28,82 %, evidenciando un uso más eficaz del contexto recuperado para generar respuestas.

Los resultados obtenidos, alineados con el objetivo del proyecto, evidencian que la incorporación del **sentimiento** como variable de entrada mejora significativamente la calidad y pertinencia de las respuestas generadas en entornos de asistencia conversacional. Esta integración permite al modelo producir respuestas más coherentes, relevantes y contextualizadas, lo que resulta especialmente valioso en escenarios donde la empatía, el tono y la adecuación semántica son fundamentales, como en los sistemas de atención al cliente.

En conclusión, se confirma que incluir explícitamente el sentimiento no solo optimiza el desempeño del modelo en métricas clave de comprensión y fidelidad, sino que también contribuye a generar respuestas más humanas y funcionales, consolidando así su valor como componente estratégico en arquitecturas de generación de lenguaje natural.

7.4.5. Conclusiones

Los resultados obtenidos permiten concluir que el modelo seleccionado **Llama2 - 7B** logró un desempeño satisfactorio en tareas de generación de respuestas en lenguaje natural, especialmente en términos de **fidelidad y relevancia**. A pesar de no alcanzar valores máximos en métricas clásicas como BLEU o ROUGE, el modelo demostró ser coherente, consistente con el contexto y adecuado para aplicaciones prácticas de atención al cliente automatizada.

Estos resultados validan el enfoque de entrenamiento eficiente mediante **LoRA + QLoRA** y confirman que es posible obtener un modelo funcional y competitivo

sin requerir infraestructura de cómputo de gran escala.

Adicionalmente, la evaluación comparativa entre los escenarios *con* y *sin* la variable **sentimiento** evidenció mejoras sustanciales en la calidad de las respuestas generadas al incorporar esta dimensión emocional. En particular, se observaron incrementos significativos en las métricas semánticas evaluadas mediante el conjunto **RAGAS**: *Faithfulness* aumentó en un 11,2 %, *Answer Relevancy* en un 2,1 % y *Context Precision* en un 28,8 %. Estos resultados respaldan la hipótesis de que integrar variables afectivas permite generar respuestas más precisas, coherentes y alineadas con el tono emocional del usuario. Esta mejora cuantitativa refuerza la decisión de enriquecer el conjunto de datos con esta variable adicional, destacando su utilidad para optimizar la interacción en entornos reales de atención al cliente.

Capítulo 8

Desarrollo de un Chatbot como Plataforma de Atención al Cliente

8.1. Implementación del Chatbot Inteligente

Como parte de los objetivos específicos del proyecto, se desarrolló un chatbot inteligente que integra los modelos previamente entrenados de análisis de sentimientos (AS) y lenguaje natural (LLM), convirtiéndose en el canal principal de comunicación entre el cliente y el sistema. Este chatbot no solo permite atender consultas de manera automatizada, sino que también personaliza las respuestas en función del estado emocional del usuario detectado por el modelo de AS, lo que mejora significativamente la calidad y efectividad de la interacción. A continuación, se describe el proceso de integración y despliegue de esta solución.

8.2. Arquitectura General del Sistema

En esta sección se describe la integración técnica de los dos modelos principales desarrollados durante el proyecto: el modelo de análisis de sentimientos y el modelo generativo basado en LLM. Ambos componentes trabajan de forma conjunta para ofrecer respuestas personalizadas, emocionalmente adecuadas y contextualizadas a las consultas de los usuarios en el ámbito del servicio al cliente.

8.2.1. Componentes Principales

El sistema ClientMinds está compuesto por tres módulos funcionales principales que trabajan de forma integrada:

- **Frontend conversacional**: Interfaz de usuario desarrollada con Streamlit, que permite la interacción en lenguaje natural, la visualización de respuestas y el seguimiento del historial.
- **Modelo de análisis de sentimientos**: Clasifica emocionalmente las instrucciones del usuario mediante un modelo BIGRU entrenado previamente.

- **Modelo de lenguaje generativo (LLM)**: Basado en LLaMA2-7B ajustado con LoRA y QLoRA, genera respuestas contextuales personalizadas según el sentimiento detectado.

8.2.2. Tecnologías y Herramientas Utilizadas

- **Streamlit**: para construir la interfaz de interacción.
- **Transformers, Peft, Accelerate (Hugging Face)**: para cargar y operar el modelo LLM ajustado.
- **TensorFlow** + **Keras**: para el modelo de análisis de sentimientos (BIGRU).
- **Googletrans** + **langdetect**: para traducir y detectar automáticamente el idioma de entrada.
- **CUDA y PyTorch**: para inferencia optimizada en GPU.

8.3. Integración de los Modelos Desarrollados

8.3.1. Modelo de Análisis de Sentimientos

El primer componente del sistema corresponde a un modelo de clasificación emocional basado en una arquitectura **BIGRU (Bidirectional Gated Recurrent Unit)**. Tal como se detalló en el Capítulo 4, este modelo fue entrenado sobre un corpus anotado de mensajes de clientes, y está diseñado para identificar el sentimiento dominante (positivo, negativo o neutral) en el texto de entrada.

Para garantizar su integración en el sistema de inferencia, el modelo fue exportado en el formato estándar `SavedModel` de TensorFlow y posteriormente cargado mediante la clase `TFSMLayer` de Keras. Dado que el modelo fue entrenado sobre datos en inglés, toda entrada en español es traducida automáticamente antes del análisis de sentimiento, asegurando la correcta interpretación del contexto emocional independientemente del idioma original del usuario.

8.3.2. Modelo LLM para Generación de Respuestas

El segundo componente clave del sistema es el modelo generativo basado en lenguaje natural. En particular, se utilizó **LLaMA2-7B**, un modelo de lenguaje de gran escala de código abierto desarrollado por Meta. Como se explicó en profundidad en el Capítulo 5, este modelo fue adaptado al dominio de atención al cliente mediante técnicas de **ajuste fino eficiente** como **LoRA (Low-Rank Adaptation)** y su extensión **QLoRA**, la cual permite entrenar modelos cuantizados en 4 bits reduciendo significativamente el consumo de memoria.

El modelo recibe como entrada un **prompt estructurado** que incluye tanto la instrucción del usuario como el sentimiento previamente detectado por el modelo

BIGRU. Esta combinación permite al modelo LLM generar una respuesta contextualizada que no solo es funcionalmente precisa, sino también empática y alineada con el estado emocional del usuario.

8.4. Flujo de Datos e Interacción con el Usuario

8.4.1. Proceso de Entrada y Clasificación Emocional

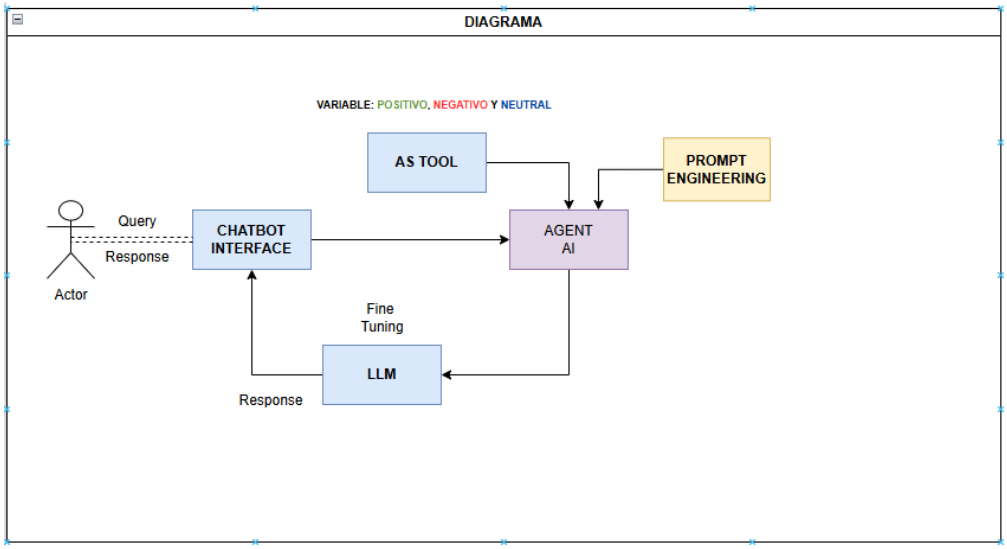


Figura 8.1: Arquitectura general del sistema de atención al cliente basada en LLMs

Descripción del Diagrama

- **Actor**: Representa al usuario final que interactúa con el sistema, realizando una consulta (*Query*) y recibiendo una respuesta.
- **Chatbot Interface**: Es la interfaz de comunicación entre el usuario y el sistema. Se encarga de recibir la consulta y mostrar la respuesta generada. Esta interfaz también puede enrutar la información hacia el **Agente AI**.
- **AS Tool (Análisis de Sentimientos)**: Procesa la consulta y clasifica la emoción o polaridad en una de tres categorías: **positiva**, **negativa** o **neutral**. Esta variable es fundamental para adaptar el tono de la respuesta del sistema.
- **Prompt Engineering**: Genera instrucciones estructuradas (prompts) considerando la consulta original y el análisis emocional realizado. Esto asegura que el modelo genere respuestas coherentes y empáticas.
- **Agente AI**: Es el componente que orquesta la operación. Recibe la entrada procesada desde el Chatbot, el análisis de sentimientos y el prompt construido, y decide cómo proceder (e.g., consulta al modelo LLM).

- **LLM (Large Language Model)**: Es el modelo de lenguaje ajustado mediante *fine-tuning*. A partir del prompt, genera una respuesta natural y adecuada, que es enviada nuevamente a la interfaz para ser mostrada al usuario.

Al ingresar una consulta, el sistema detecta automáticamente el idioma y, si es necesario, la traduce al inglés. A continuación, se aplica el modelo BIGRU para detectar el sentimiento dominante de la instrucción, el cual será utilizado para adaptar el tono de la respuesta.

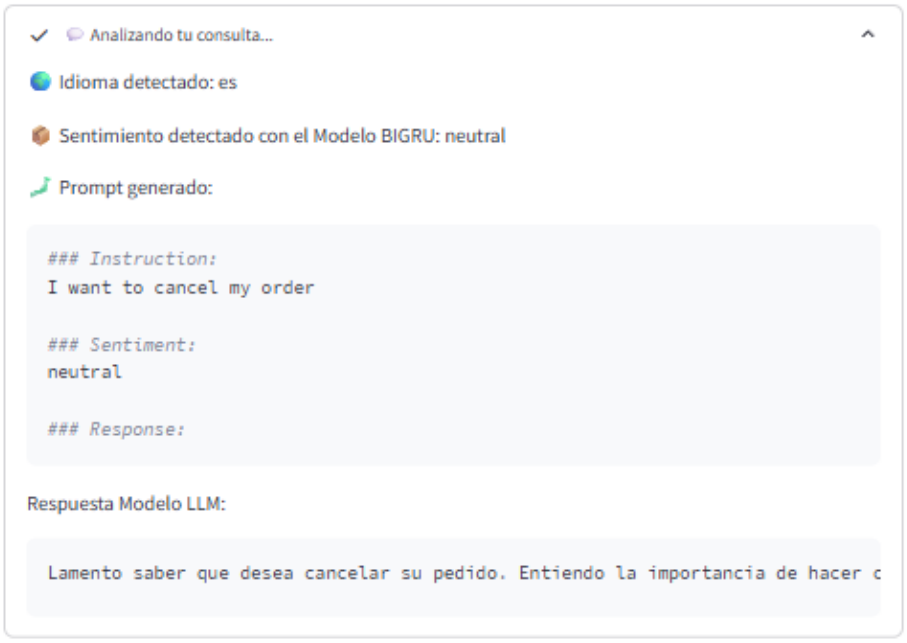


Figura 8.2: Procesamiento de informacion

8.4.2. Generación de Respuesta Personalizada

6.3.2.1 Fine-Tuning vs Prompt Engineering

Durante el desarrollo del sistema se emplearon dos enfoques complementarios para controlar el comportamiento del modelo de lenguaje: **fine-tuning** y **prompt engineering**.

6.3.2.2. Fine-Tuning:

En el proceso de ajuste fino (fine-tuning), el modelo fue entrenado con ejemplos estructurados que incluían una instrucción, un sentimiento asociado y una respuesta esperada. Un ejemplo de prompt utilizado durante este entrenamiento tenía la siguiente estructura:

```
### Instruction:
{instruccion_en}
```

```
### Sentiment:
{sentimiento_en}

### Response:
```

Esta estructura permite que el modelo aprenda a generar respuestas que consideren tanto el contenido de la instrucción como el tono emocional, permitiendo una generalización mejorada en escenarios similares.

6.3.2.3. Prompt Engineering:

En la etapa de inferencia, se aplicó *prompt engineering* para guiar al modelo con un contexto más explícito y detallado. A diferencia del fine-tuning, aquí no se modifican los pesos del modelo, sino que se proporciona un texto cuidadosamente diseñado para inducir un comportamiento deseado.

Un ejemplo de prompt usado en inferencia es el siguiente:

```
You are a professional CRM assistant. Your job is to respond...

Context:
Their sentiment has been detected as: {sentimiento}

Instructions:
- If the sentiment is Negative: respond with
empathy and a professional, helpful tone.
...
User's message:
{instruccion}

Response:
"""
```

8.4.3. Comparación y Alucinaciones:

- El **fine-tuning** permite al modelo aprender de manera estructurada a lo largo de muchos ejemplos, reduciendo la probabilidad de respuestas inconsistentes o *alucinaciones* (hallucinations). Este enfoque es ideal para capturar patrones específicos del dominio.
- El **prompt engineering** es útil cuando no se puede o no se desea ajustar los pesos del modelo. Sin embargo, su efectividad depende de la claridad y estructura del prompt. Prompts mal diseñados pueden inducir errores o generar contenido ficticio no soportado por el contexto.
- Las **alucinaciones** son respuestas generadas que suenan plausibles pero que no están basadas en la entrada o en información real. Aunque el fine-tuning

reduce su frecuencia al especializar el modelo, el uso de prompts bien definidos también puede mitigarlas significativamente durante la inferencia.

Ambos enfoques son complementarios: el fine-tuning adapta el modelo a un dominio específico, mientras que el prompt engineering permite flexibilidad y control sin necesidad de volver a entrenar el modelo.

8.4.4. Ejemplo de Interacción

La Figura 8.3 ilustra un ejemplo real de interacción con el asistente, donde se detecta un sentimiento neutral, se genera el prompt y se produce una respuesta personalizada que reconoce la intención del usuario y solicita detalles adicionales con un tono cordial.

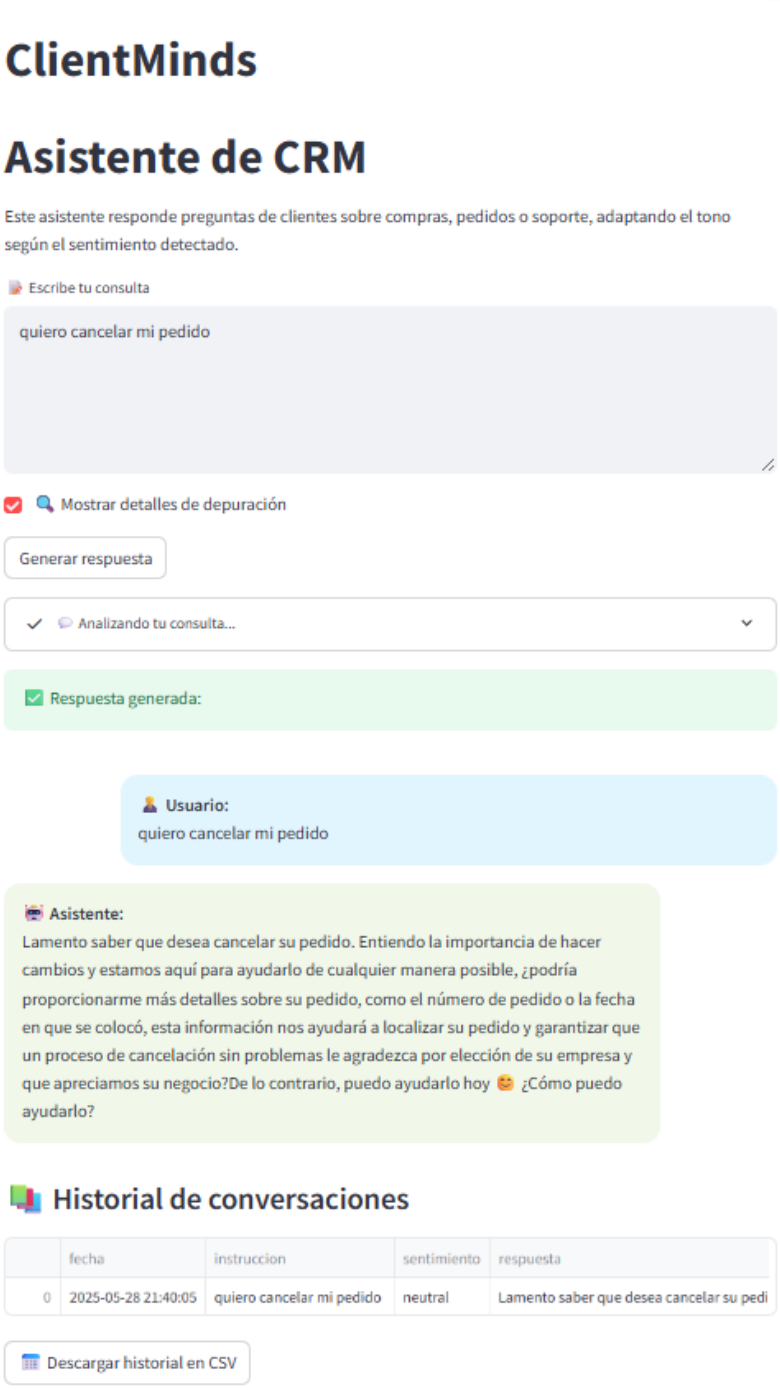


Figura 8.3: Interfaz del asistente ClientMinds con respuesta generada

8.5. Despliegue del Sistema

8.5.1. Entorno de Producción

El sistema fue desplegado localmente en una estación de trabajo con las siguientes características:

- **GPU:** NVIDIA RTX 3090 Ti (24 GB VRAM)
- **CUDA:** 12.6, cuDNN: 9.0.1
- **PyTorch:** 2.5.1
- **Sistema operativo:** Linux (Ubuntu 22.04)

8.5.2. Pruebas de Funcionamiento

Se llevaron a cabo pruebas funcionales usando instrucciones simuladas de clientes en diferentes tonos (quejas, dudas, confirmaciones). El sistema respondió correctamente, adaptando su estilo al sentimiento detectado y generando respuestas de alta coherencia y relevancia.

8.6. Resultados y Observaciones

8.6.1. Evaluación Funcional

El sistema mostró un desempeño consistente, con:

- Precisión en la detección de sentimiento.
- Reducción significativa de alucinaciones gracias a la ingeniería de prompts.
- Respuestas adaptadas a la intención y tono emocional del usuario.
- Capacidad de mantener historial y exportarlo como CSV.

8.6.2. Evaluación Cualitativa con Encuesta

Se evaluaron 10 pares de respuestas (con y sin ingeniería de prompts) mediante una encuesta aplicada a evaluadores humanos. Los resultados mostraron una mejora clara al incorporar instrucciones explícitas:

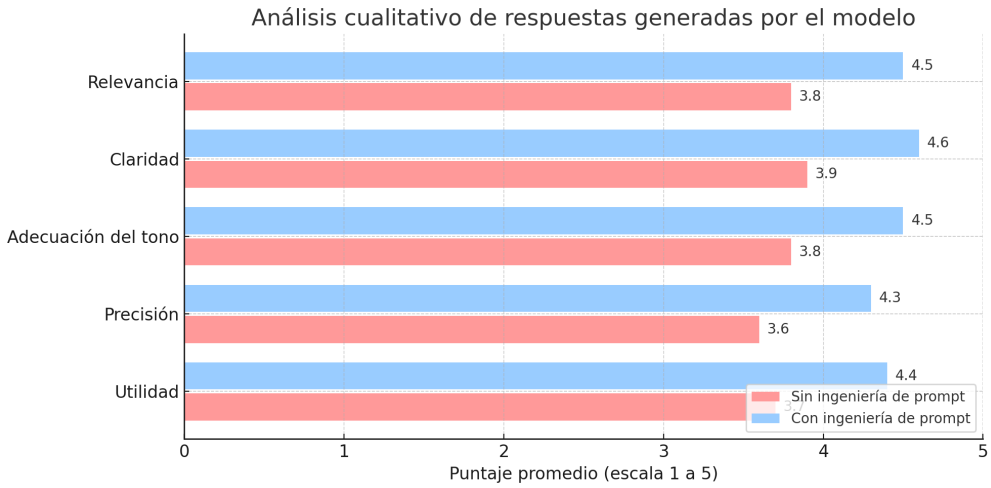


Figura 8.4: Evaluación cualitativa de respuestas generadas con y sin ingeniería de prompts

La encuesta evaluó los siguientes criterios (ver Tabla 8.1):

Criterio	**Descripción**
Relevancia	Correspondencia entre la consulta y la respuesta.
Claridad	Facilidad para entender la respuesta.
Adecuación del tono	Concordancia del tono con el sentimiento detectado.
Precisión	Corrección y coherencia de la información.
Utilidad	Valor práctico de la respuesta para el usuario.

Cuadro 8.1: Criterios evaluados en la encuesta cualitativa

Observaciones principales:

- En todos los criterios, las respuestas generadas con ingeniería de prompt obtuvieron puntajes más altos.
- Las diferencias son cercanas, pero consistentes: por ejemplo, la claridad mejoró de 3.9 a 4.6, y la utilidad de 3.7 a 4.4.
- Esto indica que aunque el modelo ya es competente con prompts simples, proporcionar instrucciones explícitas mejora su capacidad para generar respuestas más útiles, precisas y adecuadas al tono emocional del usuario.

Este resultado confirma que la ingeniería de prompts es una estrategia eficaz para refinar el comportamiento de modelos generativos sin necesidad de modificaciones adicionales en la arquitectura o reentrenamiento, y que su implementación práctica resulta beneficiosa en sistemas de atención automatizada como ClientMinds.

Capítulo 9

Reproducibilidad de los Resultados

Con el objetivo de garantizar la transparencia y la reproducibilidad del presente trabajo, se ha dispuesto un repositorio donde se encuentra el código fuente desarrollado, así como los recursos necesarios para replicar los experimentos y resultados obtenidos. El repositorio se encuentra disponible dando clic **aquí**.

9.1. Estructura del Repositorio

El repositorio está organizado en las siguientes carpetas y archivos principales:

- `/scripts/limpieza`: scripts utilizados para la limpieza y preparación del dataset.
- `/scripts/modelado`: código correspondiente a la construcción, entrenamiento y evaluación de los modelos de análisis de sentimientos y el fine tuning aplicado a los LLMs.
- `/requirements.txt`: listado de librerías y versiones necesarias para la ejecución del proyecto.
- `/README.md`: archivo con instrucciones detalladas para la replicación del entorno y ejecución de cada etapa del proyecto.

9.2. Acceso al Dataset

El dataset utilizado para el entrenamiento y evaluación de los modelos es de acceso público y se encuentra disponible en la plataforma *Hugging Face*. Puede accederse directamente dando clic **aquí**.

Se recomienda consultar el Capítulo 5 para una descripción completa del dataset, así como los pasos seguidos para su preprocesamiento y uso en los modelos propuestos.

9.3. Entorno de Ejecución

Para garantizar la compatibilidad con los scripts incluidos, se recomienda replicar el ambiente de ejecución utilizando los requerimientos especificados en el archivo `requirements.txt`. Alternativamente, es posible crear el entorno mediante el uso de entornos virtuales con `venv` o `conda`.

Ejemplo con `pip`:

```
python -m venv venv
source venv/bin/activate  # o venv\Scripts\activate en Windows
pip install -r requirements.txt
```

9.4. Consideraciones Finales

Este repositorio tiene como objetivo facilitar la replicación completa del trabajo presentado, desde la preparación del dataset hasta la evaluación de resultados. Se invita al lector a explorar, ejecutar y adaptar el código conforme a sus intereses de investigación.

Capítulo 10

Conclusiones y Trabajos Futuros

10.1. Conclusiones

1. El proyecto logró cumplir su objetivo general al desarrollar un chatbot capaz de atender consultas de usuarios mediante la integración efectiva de modelos de lenguaje de gran tamaño (LLM) y análisis de sentimientos. A lo largo del desarrollo, se vivieron las diferentes fases de un proyecto de ciencia de datos desde la recolección, exploración y limpieza de datos, hasta el modelado, evaluación e implementación, lo que permitió aplicar un enfoque riguroso y estructurado. La preparación de los datos textuales proporcionó una base sólida para entrenar los modelos; el análisis de sentimientos permitió comprender el estado emocional de los usuarios, y el fine-tuning de LLM habilitó respuestas precisas y contextualizadas. Finalmente, se integraron estos elementos en un chatbot funcional, inteligente y personalizado, que mejora significativamente la experiencia del cliente y sienta las bases para futuras mejoras basadas en analítica avanzada y automatización conversacional.

2. La evaluación comparativa entre los escenarios *con* y *sin* la variable **sentimiento** evidenció mejoras sustanciales en la calidad de las respuestas generadas al incorporar esta dimensión emocional. En particular, se observaron incrementos significativos en las métricas semánticas evaluadas mediante el conjunto **RAGAS**: *Faithfulness* aumentó en un 11,2 %, *Answer Relevancy* en un 2,1 % y *Context Precision* en un 28,8 %. Estos resultados respaldan la hipótesis de que integrar variables afectivas permite generar respuestas más precisas, coherentes y alineadas con el tono emocional del usuario. Esta mejora cuantitativa refuerza la decisión de enriquecer el conjunto de datos con esta variable adicional, destacando su utilidad para optimizar la interacción en entornos reales de atención al cliente.

3. El modelo BiGRU demostró ser la arquitectura más eficiente y robusta para la clasificación de sentimientos en el contexto de atención al cliente, logrando un equilibrio óptimo entre precisión y costo computacional, lo cual es fundamental para implementaciones prácticas en sistemas con recursos limitados.

4. La optimización rigurosa de hiperparámetros, incluyendo técnicas como EarlyStopping, fue determinante para maximizar la capacidad del modelo BiGRU de generalizar sobre datos no vistos, alcanzando un desempeño superior con un f1-score mayor a 0.92, lo que valida la importancia de un proceso de ajuste cuidadoso en proyectos de PLN.

5. La integración del análisis de sentimientos como una capa estratégica en la interpretación de interacciones con clientes permitió identificar con precisión los estados emocionales, aportando insights valiosos que pueden guiar acciones focalizadas para mejorar la experiencia y fidelización del usuario.

6. Mediante el uso de técnicas de fine-tuning basadas en LoRA y QLoRA, se logró adaptar un modelo LLaMA2-7B a las necesidades específicas del dominio sin necesidad de infraestructura de cómputo masiva, demostrando que es posible desarrollar soluciones competitivas y accesibles para generación automatizada de respuestas.

7. La evaluación integral a través del framework RAGAS evidenció que el modelo ajustado mantiene una alta fidelidad y relevancia contextual en la generación de respuestas, lo que es crucial para aplicaciones de atención al cliente donde la coherencia y la precisión de la información son requisitos indispensables.

8. A pesar de limitaciones en métricas clásicas de generación como BLEU y ROUGE, los resultados sugieren que la variabilidad inherente del lenguaje natural y la flexibilidad en la formulación de respuestas pueden ser abordadas con modelos LLM bien ajustados, permitiendo un diálogo más natural y efectivo con los usuarios.

9. La creación del chatbot integrado con modelos de análisis de sentimientos y LLM permitió una atención personalizada, efectiva y contextualizada, constituyendo una solución escalable para la automatización de consultas en sistemas de atención al cliente.

10. Se identificaron limitaciones operativas relevantes, como la ausencia de memoria conversacional y la dependencia de sistemas de traducción, que impactan la continuidad y naturalidad del diálogo, planteando un área crítica para mejoras en futuras versiones.

11. La incorporación futura de técnicas avanzadas como recuperación de contexto (RAG) y segmentación por intenciones promete ampliar sustancialmente la capacidad del chatbot para manejar interacciones complejas y multi-turno, alineándose con las mejores prácticas actuales en PLN.

12. En conjunto, el trabajo evidencia que la combinación de análisis profundo de datos textuales, optimización de modelos de sentimientos y ajuste fino de LLM constituye una estrategia efectiva para desarrollar sistemas conversacionales inteligentes que mejoran la experiencia del usuario y potencian la relación cliente-marca.

10.2. Trabajos Futuros

El desarrollo del chatbot presentado en este proyecto no constituye un punto final, sino una base sólida sobre la cual se pueden construir nuevas capacidades que fortalezcan la experiencia del cliente a través de soluciones cada vez más inteligentes y adaptativas. A partir de los resultados obtenidos y de las limitaciones identificadas, se plantean a continuación tres líneas de trabajo futuro que permitirán continuar la evolución de este sistema conversacional.

1. Evolución hacia agentes reactivos y proactivos

Actualmente, el chatbot funciona como un agente principalmente reactivo, es decir, responde a las solicitudes del usuario sin iniciativa propia. Un primer paso evolutivo sería convertirlo en un **agente de IA reactivo avanzado**, capaz de mantener contexto conversacional, gestionar múltiples actividades y adaptar sus respuestas en función del historial inmediato de la conversación. Más adelante, se podrá avanzar hacia un **agente proactivo**, que no solo responda sino que anticipe necesidades, sugiera acciones, y emita alertas o recomendaciones basadas en patrones de comportamiento detectados. Esto permitiría una interacción más rica, personalizada y orientada a la acción.

2. Alimentación continua y dinámica de la base de conocimiento

Para lograr un sistema verdaderamente inteligente, es necesario que la base de datos que alimenta al modelo se actualice continuamente con las nuevas interacciones generadas por los usuarios. Este mecanismo de aprendizaje dinámico permitiría refinar las predicciones y mejorar la capacidad del modelo para adaptarse a cambios en el lenguaje, nuevas consultas y comportamientos emergentes. La actualización periódica o incluso en tiempo real de los datos garantizaría una evolución constante del sistema sin necesidad de reiniciar todo el proceso de entrenamiento desde cero.

3. Explotación analítica de las interacciones registradas

Además de alimentar al modelo, las interacciones registradas representan una fuente de datos rica para realizar analítica avanzada. Estos datos pueden utilizarse para identificar tendencias, detectar puntos de dolor recurrentes, segmentar usuarios según comportamiento o sentimientos, y generar reportes que sirvan de apoyo a la toma de decisiones estratégicas. De este modo, el sistema no solo sería una herramienta operativa, sino también un activo analítico que aporte valor agregado a la organización desde una perspectiva de inteligencia de negocio.

Conclusión

Estas líneas de trabajo proponen una evolución natural del sistema desarrollado, en la que el chatbot transita desde ser una herramienta de respuesta puntual

hacia convertirse en un agente inteligente, proactivo y analítico. El enfoque planteado refuerza la idea de que la ciencia de datos, aplicada con intención y sostenibilidad, puede ser una aliada fundamental para optimizar la relación con los clientes y mejorar continuamente su experiencia a través de sistemas conversacionales cada vez más sofisticados.

Bibliografía

Referencias

[1] J. Smith y J. Jones, *Customer Relationship Management in the Modern Era*. New York, NY: Business Insights, 2022.

[2] H. A. Al-Homery y H. Ashari, «Customer Relationship Management: A Literature Review Approach,» *International Journal of Global Optimization and Its Application*, vol. 2, n.º 1, págs. 20-38, mar. de 2023, License CC BY 4.0. DOI: 10.56225/ijgoia.v2i1.160.

[3] J. Doe, «Challenges in Extracting Meaningful Insights from CRM Data,» *Journal of Business Analytics*, vol. 15, n.º 2, págs. 123-135, 2021.

[4] M. Brown y S. White, «Hidden Patterns in Customer Data,» *International Journal of Data Science*, vol. 10, n.º 4, págs. 456-470, 2019.

[5] L. Taylor y D. Green, «Responding to Customer Needs in Real Time,» *Customer Experience Review*, vol. 8, n.º 1, págs. 78-85, 2018.

[6] E. Johnson, *Data-Driven Customer Retention Strategies*. Los Angeles, CA: Marketing Press, 2020.

[7] E. G. Tavira y E. M. R. Estrada, «Marketing relacional: valor, satisfacción, lealtad y retención del cliente. análisis y reflexión teórica,» *Ciencia y Sociedad*, 2015, [En línea]. Disponible en: https://www.redalyc.org/articulo.oa?id=87041161004. [Accedido: Jun. 09, 2024].

[8] S. Lamrhari, H. E. Ghazi, M. Oubrich y A. E. Faker, «A social CRM analytic framework for improving customer retention, acquisition, and conversion,» *Technological Forecasting and Social Change*, vol. 174, pág. 121 275, 2022, [Online]. Available: https://doi.org/10.1016/j.techfore.2021.121275. [Accessed: Jun. 9, 2024].

[9] C. Ledro, A. Nosella e I. D. Pozza, «Integration of AI in CRM: Challenges and guidelines,» *J. Open Innov. Technol. Market. Complexity*, vol. 9, n.º 4, pág. 100 151, 2023, [Online]. Available: https://doi.org/10.1016/j.joitmc.2023.100151. [Accessed: Jun. 09, 2024].

[10] M. Liu, H. Zhang, Z. Xu y K. Ding, «The fusion of fuzzy theories and natural language processing: A state-of-the-art survey,» *Appl. Soft Comput.*, vol. 162, pág. 111 818, 2024, [Online]. Available: https://doi.org/10.1016/j.asoc.2024.111818.

[11] B. Liu, *Sentiment Analysis and Opinion Mining* (Synthesis Lectures on Human Language Technologies 1). Morgan & Claypool Publishers, 2012, vol. 5, págs. 1-167.

[12] W. Medhat, A. Hassan y H. Korashy, «Sentiment analysis algorithms and applications: A survey,» *Ain Shams Engineering Journal*, vol. 5, n.º 4, págs. 1093-1113, 2014.

[13] M. I. Jordan y T. M. Mitchell, «Machine learning: Trends, perspectives, and prospects,» *Science*, vol. 349, n.º 6245, págs. 255-260, 2015.

[14] S. Hochreiter y J. Schmidhuber, «Long short-term memory,» *Neural Computation*, vol. 9, n.º 8, págs. 1735-1780, 1997.

[15] K. Cho, B. van Merrienboer, C. Gulcehre et al., «Learning phrase representations using RNN encoder-decoder for statistical machine translation,» *arXiv preprint arXiv:1406.1078*, 2014.

[16] N. T. K. Le, N. Hadiprodjo, H. El-Alfy, A. Kerimzhanov y A. Teshebaev, «The Recent Large Language Models in NLP,» *2023 22nd International Symposium on Communications and Information Technologies (ISCIT)*, 2023, DOI: 10.1109/ISCIT57293.2023.10376050.

[17] A. Vaswani, N. Shazeer, N. Parmar et al., «Attention is all you need,» en *Advances in neural information processing systems*, vol. 30, 2017.

[18] J. Li y H. Yin, «Fine-tuning large pre-trained models: A survey,» *Journal of Artificial Intelligence Research*, vol. 70, págs. 1-45, 2021.

[19] J. Howard y S. Ruder, «Universal language model fine-tuning for text classification,» en *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2018, págs. 328-339.

[20] E. Hu, Y. Shen, P. Wallis et al., «LoRA: Low-Rank Adaptation of Large Language Models,» *arXiv preprint arXiv:2106.09685*, 2021.

[21] T. Dettmers, A. Pagnoni, A. Holtzman y L. Zettlemoyer, «QLoRA: Efficient Finetuning of Quantized LLMs,» *arXiv preprint arXiv:2305.14314*, 2023.

[22] A. Singhal, «Modern Information Retrieval: A Brief Overview,» *IEEE Data Engineering Bulletin*, vol. 24, n.º 4, págs. 35-43, 2001.

[23] N. Reimers e I. Gurevych, «Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks,» en *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*, Association for Computational Linguistics, 2019, págs. 3982-3992.

[24] J. Johnson, M. Douze y H. Jégou, «Billion-scale similarity search with GPUs,» *IEEE Transactions on Big Data*, vol. 7, n.º 3, págs. 535-547, 2019.

[25] R. Team, *RAGAS: Retrieval-Augmented Generation Assessment Scores*, `https://docs.ragas.io/en/stable/concepts/metrics/`, Accedido el 28 de mayo de 2025, 2023.

[26] Y. Zhang, R. Y. Lau, J. D. Xu, Y. Rao e Y. Li, «Business chatbots with deep learning technologies: state-of-the-art, taxonomies, and future research directions,» *Artificial Intelligence Review*, 2024.

[27] K. L. Ehsani, E. Rahman Rhythm, M. H. K. Mehedi y A. A. Rasel, «A Comparative Analysis of Customer Service Chatbots: Efficiency, Usability and Application,» en *2023 Computer Applications & Technological Solutions (CATS)*, IEEE, 2023. DOI: 10.1109/CATS58046.2023.10424303.

[28] Anonymous, *Impact of Chatbot Service on Bank Performance Based on a Case of IBM Watson Assistant*, online, 2024.

[29] Y. Huang, J. Chia, H. Liu et al., «CRMArena: A Benchmark for CRM Agents,» *arXiv preprint arXiv:2411.02305*, 2024.

[30] J. Kaur y B. K. Sidhu, «Sentiment Analysis Based on Deep Learning Approaches,» *2018 Second International Conference on Intelligent Computing and Control Systems (ICICCS)*, págs. 1496-1500, 2018, doi: 10.1109/ICCONS.2018.8662899.

[31] M. T. F. A. Islami, A. R. Barakbah y T. Harsono, «Interactive Applied Graph Chatbot with Semantic Recognition,» *2020 International Electronics Symposium (IES)*, págs. 557-564, 2020, doi: 10.1109/IES50839.2020.9231678.

[32] I. Goodfellow, Y. Bengio y A. Courville, *Deep Learning*. MIT Press, 2016.

[33] Y. Bengio, P. Simard y P. Frasconi, «Learning long-term dependencies with gradient descent is difficult,» *IEEE Transactions on Neural Networks*, vol. 5, n.º 2, págs. 157-166, 1994.

[34] H. Touvron, L. Martin, K. Stone, M. Cord y H. Jégou, «LLaMA: Open and Efficient Foundation Language Models,» *arXiv preprint arXiv:2307.09288*, 2023.

[35] C. Xu, D. Garcia y S. Rodriguez, «PHI: Personalized Human-Interactive Model for Customer Support,» *Proceedings of the Association for Computational Linguistics*, 2023.

[36] K. Papineni, S. Roukos, T. Ward y W. Zhu, «BLEU: a method for automatic evaluation of machine translation,» *Proceedings of the 40th annual meeting on association for computational linguistics*, págs. 311-318, 2002.

[37] C. Y. Lin, «ROUGE: A package for automatic evaluation of summaries,» *Text Summarization Branches Out: Proceedings of the ACL-04 Workshop*, págs. 74-81, 2004.

[38] J. T. Goodman, «A bit of progress in language modeling,» *Computer Speech & Language*, vol. 15, n.º 4, págs. 403-434, 2001.