



Pontificia Universidad
JAVERIANA
Cali

MODELO PARA LA ORIENTACIÓN SOBRE TRÁMITES AL CIUDADANO EN EL DISTRITO DE SANTIAGO DE CALI

GIOVANNI SÁNCHEZ SOTO
Código 8986548

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

Director DIEGO
LINARES

Codirectora
GLORIA ÁLVAREZ

FACULTAD DE INGENIERÍA Y CIENCIAS MAESTRÍA EN
CIENCIA DE DATOS SANTIAGO DE CALI, JUNIO 4 DE 2025

FICHA RESUMEN PROYECTO DE TRABAJO DE GRADO

TÍTULO DEL PROYECTO: Modelo para la orientación sobre trámites al ciudadano en el Distrito de Santiago de Cali

1. **ÁREA DE TRABAJO:** Servicio al cliente
2. **TIPO DE PROYECTO (Aplicado, Innovación, Investigación):** Aplicado
3. **ESTUDIANTE(S):** Giovanni Sánchez Soto
4. **CORREO ELECTRÓNICO:** giosan8918@javerianacali.edu.co,
5. **DIRECCIÓN Y TELEFONO:** calle 13a1 # 70-61, Cali, 3186008569
6. **DIRECTOR:** Diego Linares, Gloria Álvarez
7. **VINCULACIÓN DEL DIRECTOR:** Profesor de tiempo completo Facultad Ingeniería y Ciencias
8. **CORREO ELECTRÓNICO DEL DIRECTOR:** dlinares@javerianacali.edu.co
9. **CO-DIRECTOR (Si aplica):** galvarez@javerianacali.edu.co
10. **GRUPO O EMPRESA QUE LO AVALA (Si aplica):** Alcaldía de Santiago de Cali
11. **OTROS GRUPOS O EMPRESAS:** No aplica
12. **PALABRAS CLAVE (al menos 5):** Trámites, Servicios, Cali, Datos, Aprendizaje Automático, consulta, ciudadanas.
13. **FECHA DE INICIO:** junio de 2024- febrero de 2025
14. **DURACIÓN ESTIMADA (En meses):** 9 Meses

15. RESUMEN:

Este proyecto aplicado aborda la problemática de la atención a consultas de orientación sobre trámites al ciudadano en el Distrito de Santiago de Cali, explorando soluciones a partir del ajuste fino de Modelos de Lenguaje Grande (LLMs) para automatizar la generación de respuestas, se entrenaron modelos cuantizados Llama 3.2 3B Instruct y Phi 3.5 Mini Instruct con conjuntos de datos extraídos de plataformas oficiales del Distrito, también se implementó un sistema de Generación Aumentada por Recuperación (RAG), utilizando el modelo Llama 3.2 para la generación de respuestas. Los resultados de la evaluación demostraron que, si bien los modelos ajustados mostraron un buen desempeño semántico, presentaron problemas de factualidad mientras el enfoque basado en RAG obtuvo los mejores resultados en ambos casos.

Index Terms— Administración pública, generación aumentada por recuperación, RAG, modelos de lenguaje grande, Fine-Tuning, procesamiento de lenguaje natural, servicio al ciudadano.

Contenido

1. INTRODUCCIÓN	6
2. DEFINICIÓN DEL PROBLEMA	7
2.1 Planteamiento del problema.....	7
2.2 Formulación del problema	8
3. OBJETIVOS DEL PROYECTO	9
3.1 Objetivo general	9
3.2 Objetivos específicos	9
4. MARCO DE REFERENCIA.....	10
4.1 Definición de trámites en el Distrito de Santiago de Cali.....	10
4.2 Modelos de lenguaje grande	11
4.3 Entrenamiento de Modelos de Lenguaje Grande	12
4.3.1 Ajuste fino (fine-tuning)	13
4.4 Retrieval-Augmented Generation (RAG).....	14
4.5 Etiquetado de datos.....	15
4.6 Métricas de evaluación para modelos de lenguaje grande	16
4.6.1 Métricas automáticas	16
4.6.2 Evaluación Humana.....	18
4.7 Antecedentes.....	18
5. ENTENDIMIENTO Y PREPARACIÓN DE LOS DATOS	20
5.1 Fuentes de datos disponibles.....	20
5.2 Descripción de los datos del reporte de información general de trámites.....	26
5.3 Preprocesamiento de los datos	29
6. SELECCIÓN DE MODELOS DE LENGUAJE GRANDE.....	31
6.1 Definición de criterios de selección.....	31
6.2 Aplicación de criterios de selección	33
7. MODELADO CON MODELOS DE LENGUAJE GRANDE.....	38
7.1 Entrenamiento de modelos seleccionados	38
7.2 Evaluación de modelos entrenados.....	41
7.2.1 Generación de respuestas a partir de los modelos entrenados	41
7.2.2 Evaluación cuantitativa con BERTSCORE	42
8. MODELADO CON RAG.....	44
8.1 Aplicación de Retrieval-Augmented Generation (RAG).....	44
8.2 Resultados de la evaluación.....	45
8.3 Evaluación comparativa.....	46
8.3.1 Evaluación cuantitativa consolidada.....	46
8.3.2 Evaluación humana.....	47
8.4 Análisis de resultados	48
9. CONCLUSIONES Y TRABAJOS FUTUROS.....	50
9.1 Conclusiones.....	50
9.2 Trabajos futuros.....	51

10. ANEXOS.....	52
11. REFERENCIAS	56

Lista de Figuras

Fig.1. Atributos de los trámites.	10
Fig.2. Pasos para realizar un ajuste fino.....	14
Fig.3. Flujo de RAG.	15
Fig.4. Porcentaje de trámites sin pasos.....	26
Fig.5. Porcentaje de trámites sin hoja de vida	26
Fig.6. Porcentaje de trámites en línea y con enlace disponible.....	27
Fig. 7. Porcentaje de trámites con enlace disponible	27
Fig.8. Porcentaje de trámites con costo asociado	28
Fig.9. Gráfica de pérdida para modelo Llama 3.2 3B	40
Fig. 10. Gráfica de pérdida para modelo Phi 3.5.....	41

Lista de Tablas

TABLA 1. COMPARATIVO DE MÉTODOS DE EVALUACIÓN AUTOMÁTICA	17
TABLA 2 INFO GENERAL DESCRIPCIÓN.....	21
TABLA 3 COSTOS.....	22
TABLA 4. VERIFICACIONES.....	23
TABLA 5 PASOS PARA REALIZAR TRÁMITES	24
TABLA 6 ESTAMPILLAS.....	25
TABLA 7 EJEMPLO DE PREGUNTAS Y RESPUESTAS	29
TABLA 8. COMPARATIVO ENTRE MODELOS DE LENGUAJE GRANDE DE CÓDIGO ABIERTO	32
TABLA 9. CRITERIOS DE SELECCIÓN APLICADOS A MODELOS PRESELECCIONADOS	36
TABLA 10 HIPERPARAMETROS DEFINIDOS PARA MODELOS CON MEJOR DESEMPEÑO.....	39
TABLA 11. RESULTADOS DE MODELOS UTILIZADOS	46
TABLA 12 RESULTADOS DE EVALUACIÓN POR EXPERTOS	48

1.INTRODUCCIÓN

El Distrito de Santiago de Cali enfrenta desafíos en la gestión eficiente de trámites y servicios, el servicio al ciudadano se realiza a través de múltiples canales, tanto presenciales como virtuales, y aunque estos esfuerzos han mejorado la accesibilidad, persisten problemas críticos, los cuales se logran apreciar en el informe de seguimiento del servicio al ciudadano del primer trimestre de 2024 en el cual se reveló que solo el 54% de las solicitudes se atienden dentro del tiempo estipulado, lo que muestra una necesidad urgente de mejora de estos procesos. También se identificó que los factores que contribuyen a este bajo rendimiento son principalmente la insuficiencia de recursos humanos, la incorrecta clasificación y direccionamiento de las solicitudes, así como la alta demanda de solicitudes, estos problemas además de generar insatisfacción tienen implicaciones legales y económicas para la administración local.

Este proyecto aplicado propone el entrenamiento de Modelos de Lenguaje Grande (LLMs) con el objetivo de automatizar la generación de respuestas a consultas de orientación relacionadas con trámites, para ello se entrenaron modelos cuantizados de Llama 3.2 3B y Phi 3.5 mini a partir de diferentes fuentes de datos disponibles en las plataformas de gestión de trámites que permitieron al modelo aprender en la generación de respuestas.

Este proyecto tiene el potencial de mejorar significativamente la interacción entre los ciudadanos y la administración pública en Cali, dando un paso inicial hacia la mejora de la eficiencia del servicio, en donde la implementación del modelo si bien no hace parte del presente proyecto, sirve como base para el desarrollo de un nuevo canal de atención en la administración pública.

2. DEFINICIÓN DEL PROBLEMA

2.1 Planteamiento del problema

El servicio al ciudadano en la Alcaldía Distrital de Santiago de Cali se realiza mediante diferentes medios, como la ventanilla única donde los ciudadanos acuden presencialmente para solicitar orientación o radicar peticiones, quejas y reclamos, también a través de medios virtuales como el correo electrónico, la página web de la entidad o el canal telefónico. Del informe de seguimiento de servicio al ciudadano del primer trimestre del 2024 se pueden evidenciar aspectos a mejorar, entre los cuales se destaca que las respuestas no se realizan dentro de los tiempos estipulados para la generalidad de las solicitudes recibidas, se observa que el indicador de eficiencia cuyo objetivo es medir el porcentaje de comunicaciones atendidas a tiempo alcanza solo el 54% en su última medición [1], este bajo desempeño que además es recurrente puede deberse a múltiples causas, dentro de las cuales se identifican la falta de recursos humanos necesarios para dar respuesta al número de solicitudes presentadas, por otra parte, es bien sabido que su correcta clasificación y direccionamiento tiene un impacto considerable en los tiempos de respuesta, también que el desconocimiento de los ciudadanos respecto a cómo realizar algún trámite hace que se ocupe la capacidad del personal disponible para atender estas necesidades, esto es evidente teniendo en cuenta que el número de solicitudes de orientación que se registran a través de las encuestas de satisfacción en los puntos de atención presenciales ocupan el tercer lugar de todas las gestiones realizadas con un 11,8%, siendo las solicitudes de trámites y servicios las que se encuentran en el primer lugar con 65,9% y el registro de PQRSD en el segundo con 17,2%.

Otro aspecto a tener en cuenta corresponde a la inconformidad con el servicio, en el cual aquellos que generan mayor insatisfacción para los trámites y servicios en línea son la 'Disponibilidad de la página' con 31%, seguido por 'No obtuvo el trámite' con 21, por otro lado, si bien algunos trámites no tienen costo, es bien sabido que existen personas que cobran por ellos haciéndose pasar por funcionarios o como intermediarios, de allí que la entidad destine recursos para comunicar a la población sobre cómo realizar algunos trámites y así ayudar a prevenir este tipo de prácticas. Esta situación además de generar insatisfacción entre los usuarios que utilizan los medios de atención que tiene disponible el Distrito, tiene implicaciones legales que pueden tener impactos económicos y de imagen en cuanto existan incumplimientos en los tiempos dispuestos por la ley para atender las solicitudes de los ciudadanos.

Frente a los trámites, la Alcaldía de Cali usa dos plataformas en paralelo para registrar la hoja de vida que los caracteriza, una es la plataforma oficial de la nación SUIT administrada por el Departamento Administrativo de la Función Pública y otra administrada directamente por la Alcaldía de Cali denominada Sistema de Gestión de Trámites (SGT). Con respecto a los servicios en la plataforma SGT se registran algunas características y otros elementos se encuentran dispersos a través de la página web oficial de la Alcaldía, así como de documentos que establecen las características de algunos de los programas de la entidad, esto hace que la consulta de la información sea una dificultad que enfrentan a diario los funcionarios que atienden al público, quienes recurren a hojas de cálculo o consultas a otros organismos para lograr responder a las solicitudes de orientación de los ciudadanos, otro elemento a considerar corresponde a la posibilidad de brindar información errónea o desactualizada lo que ocurre con frecuencia.

Desde la perspectiva de la ciencia de datos se logran evidenciar retos técnicos en cuanto a la consolidación de los datos para su procesamiento de tal forma que se puedan atender desde el entrenamiento de un

modelo, buscando una alternativa que a través de un modelo pueda dar respuesta automatizada a las solicitudes de orientación de los ciudadanos relacionadas con el eje temático de trámites.

2.2 Formulación del problema

Teniendo en cuenta el problema planteado se buscó responder a los siguientes interrogantes: ¿Se puede responder automáticamente a consultas relacionadas con la orientación frente a trámites y servicios del Distrito de Santiago de Cali mediante el entrenamiento de un modelo de lenguaje Grande LLM?

Sistematización:

- ¿Cuál modelo de lenguaje grande LLM se adapta mejor a las necesidades y recursos disponibles para lograr su entrenamiento?
- ¿Cómo deberían recolectarse y procesarse los datos disponibles para entrenar el modelo?
- ¿Cómo debería entrenarse el modelo LLM para dar respuesta a las consultas realizadas?
- ¿Cuáles serán las métricas más adecuadas para evaluar el desempeño del modelo?

3. OBJETIVOS DEL PROYECTO

3.1 Objetivo general

Entrenar un modelo de lenguaje grande LLM para que logre dar respuesta a consultas relacionadas con la orientación frente a trámites del Distrito de Santiago de Cali.

3.2 Objetivos específicos

- Determinar cuál es el LLM que mejor se adapta a las necesidades y recursos disponibles para lograr su entrenamiento.
- Recolectar y procesar los datos disponibles relacionados con trámites en el Distrito de Santiago de Cali y almacenarlos en una base de datos para el entrenamiento del modelo.
- Entrenar el modelo utilizando los datos procesados y técnicas de procesamiento de lenguaje natural, para que logre responder a las consultas de orientación relacionadas con trámites del Distrito de Santiago de Cali.
- Evaluar el desempeño del modelo para dar respuesta a las consultas realizadas.

4. MARCO DE REFERENCIA

4.1 Definición de trámites en el Distrito de Santiago de Cali

El Departamento Administrativo de la Función Pública (DAFP) en Colombia es un organismo de la rama ejecutiva del orden nacional que tiene dentro de sus funciones gestionar los diferentes sistemas de información que permitan el seguimiento, análisis y evaluación del desempeño de la administración pública y la toma de decisiones para una mejor prestación del servicio público, en el marco de sus funciones ha implementado el Sistema Único de Trámites (SUIT) y dispuesto los conceptos relativos a la caracterización de los trámites [2].

1) *Sistema Único de Información de Trámites (SUIT)*: Es el instrumento de acceso a la información de los trámites y otros procedimientos administrativos del Estado. Es la única fuente válida de información exigible y oponible a los ciudadanos, usuarios y grupos de interés. Así mismo, es la herramienta que facilita la implementación de la política de racionalización de trámites establecida en la Ley 962 de 2005 y en el Decreto Ley 019 de 2012 [3].

2) *Trámite*: De acuerdo con el DAFP un trámite es un conjunto de requisitos, pasos o acciones, regulados por el estado dentro de un procedimiento administrativo misional que deben efectuar los ciudadanos ante una institución de la administración pública, o particular que ejerce funciones administrativas, para hacer efectivo un derecho o cumplir con una obligación prevista o autorizada por la ley, cuyo resultado es un producto o servicio [3], en la figura 1 se logran apreciar los diferentes atributos de los trámites.



Fig. 1. Atributos de los trámites [4].

4.2 Modelos de lenguaje grande

Procesamiento de lenguaje natural NLP: El Procesamiento de Lenguaje Natural (NLP, por sus siglas en inglés) es una rama de la inteligencia artificial que se centra en la interacción entre las computadoras y el lenguaje humano, el objetivo de NLP es permitir a las máquinas entender, interpretar y generar lenguaje humano de una manera que sea tanto valiosa como significativa.

El procesamiento de lenguaje natural combina conocimientos de lingüística, ciencias de la computación e inteligencia artificial, una de sus bases teóricas más importantes es la teoría de Chomsky sobre las estructuras sintácticas, que sugiere que el lenguaje humano tiene una estructura subyacente que puede ser descrita formalmente, esta teoría ha sido fundamental para el desarrollo de gramáticas formales, que son la base de los algoritmos de procesamiento de lenguaje natural (NLP) utilizados para analizar y generar lenguaje, estos algoritmos requieren tres componentes clave: análisis morfológico, sintáctico y semántico.

a) Análisis morfológico: descompone las palabras en sus morfemas, las unidades mínimas de significado.

b) Análisis sintáctico: examina la estructura gramatical de las frases.

c) Análisis semántico: intenta comprender el significado de las palabras y frases en un contexto específico.

Estos procesos son esenciales para que las máquinas puedan interpretar y procesar el lenguaje humano con precisión.[5].

Inicialmente, los modelos de NLP (Procesamiento de Lenguaje Natural) se basaban en reglas explícitas codificadas por expertos humanos, sin embargo, con el tiempo, estos modelos han sido reemplazados en gran medida por modelos estadísticos y de aprendizaje automático, los modelos estadísticos utilizan grandes corpus de texto para identificar patrones y probabilidades asociadas con diferentes estructuras de lenguaje, el modelo de Márkov oculto y los árboles de decisión son ejemplos de tales enfoques que han sido utilizados para tareas como el etiquetado de partes de la oración y el análisis sintáctico [6]. En la última década, los modelos de aprendizaje profundo, particularmente las redes neuronales profundas, han revolucionado el campo del NLP, estos modelos se basan en arquitecturas complejas de redes neuronales que pueden aprender representaciones de alto nivel del lenguaje a partir de grandes cantidades de datos sin supervisión explícita, el uso de redes neuronales recurrentes (RNN) y redes neuronales convolucionales (CNN) ha permitido avances significativos en tareas como la traducción automática, el resumen de textos y la generación de lenguaje natural [8].

Siendo uno de los desarrollos más significativos la introducción de los modelos de lenguaje grande que han superado a las RNN en varias tareas de NLP, los transformadores utilizan mecanismos de atención para procesar todas las palabras de una secuencia en paralelo, lo que les permite manejar dependencias de largo alcance en el texto de manera más eficiente, modelos como BERT (Bidirectional Encoder Representations from Transformers) de Google y GPT (Generative Pre-trained Transformer) de Open AI han establecido nuevos estándares en diversas tareas de NLP, desde la comprensión de texto hasta la generación de lenguaje, estos modelos se entrenan inicialmente en un corpus masivo y variado de texto sin ninguna etiqueta específica, usando objetivos de aprendizaje no supervisado como la predicción de palabras enmascaradas, para reentrenarlos se usan técnicas como el ajuste fino usando un conjunto de datos más pequeño y específico con etiquetas y ajustando los hiperparámetros del modelo para que se desempeñen mejor en una tarea particular [9].

Estos operan bajo el principio central de predecir la siguiente palabra (o token) en una secuencia, dadas las palabras precedentes, a través del modelado estadístico del lenguaje, donde la probabilidad de que una palabra aparezca en un contexto determinado se aprende de los datos de entrenamiento[6]. Un concepto clave en este proceso es la conversión de datos a un formato vectorial, conocido como embedding, para su procesamiento por la red neuronal[7]. El éxito de los LLMs modernos se atribuye en gran medida a la arquitectura Transformer. Sus componentes clave incluyen:

1)*Mecanismo de Autoatención (Self-Attention)*: Permite al modelo ponderar la importancia de diferentes palabras en la secuencia de entrada al procesar cada palabra, esto es fundamental para capturar dependencias a largo alcance de manera efectiva.

2)*Procesamiento Paralelo*: A diferencia de las Redes Neuronales Recurrentes (RNNs), los Transformers pueden procesar todas las palabras de la secuencia de entrada en paralelo. Esto acelera significativamente el entrenamiento, especialmente con el uso de GPUs.

3)*Estructura Codificador-Decodificador vs. Solo Decodificador*: Aunque el Transformer original tenía una estructura de codificador para entender la entrada y decodificador para generar la salida, las arquitecturas de solo decodificador son prevalentes para los LLMs generativos actuales.

4) *Codificación Posicional (Positional Encoding)*: Dado que el procesamiento está en paralelo y se pierde la información de orden secuencial inherente, se utiliza la codificación posicional para informar al modelo sobre la posición de las palabras en la secuencia.

5) *La arquitectura Transformer representa un compromiso de diseño*: la autoatención y el procesamiento paralelo permiten escalar a conjuntos de datos masivos, pero introducen desafíos como la complejidad cuadrática del mecanismo de atención con respecto a la longitud de la secuencia y la necesidad de componentes específicos como la codificación posicional. La capacidad de procesar tokens en paralelo es clave para el entrenamiento a gran escala, pero elimina la información secuencial intrínseca, requiriendo codificaciones posicionales [8].

4.3 Entrenamiento de Modelos de Lenguaje Grande

La escala es un factor crítico en el rendimiento y las capacidades de los LLMs, abarcando el tamaño del modelo (número de parámetros), el tamaño del conjunto de datos de entrenamiento y la potencia computacional requerida, los LLMs modernos contienen decenas o cientos de miles de millones de parámetros y se entrenan en conjuntos de datos con billones de tokens.

Un fenómeno notable es la aparición de habilidades emergentes en modelos muy grandes, capacidades que no están presentes o no son evidentes en modelos más pequeños, estas incluyen el aprendizaje en contexto (aprender de ejemplos proporcionados en el prompt), el seguimiento de instrucciones y el razonamiento multi-paso. Raschka menciona este comportamiento emergente [7]. Las Leyes de Escalamiento (Scaling Laws), como las propuestas por Kaplan et al. [10] y Hoffmann et al. [11], predicen que a medida que aumentan el tamaño del modelo y los datos de entrenamiento, el rendimiento de los LLMs en diversas tareas lingüísticas continúa mejorando.

El aumento de escala en los LLMs no solo conduce a mejoras cuantitativas, sino también a cambios cualitativos en sus capacidades, mientras que los modelos más pequeños pueden funcionar bien en tareas específicas para las que fueron entrenados, fenómenos como el aprendizaje few-shot, el seguimiento complejo de instrucciones y el razonamiento multi-paso solo aparecen de manera fiable una vez que los modelos alcanzan un cierto umbral de parámetros y datos de entrenamiento, lo cual sugiere que la escala

permite a los modelos aprender principios subyacentes más generales del lenguaje y el razonamiento, no sólo patrones superficiales, cambiando fundamentalmente lo que pueden hacer, no sólo cuán bien lo hacen.

4.3.1 Ajuste fino (fine-tuning)

El ajuste fino o fine-tuning es el proceso de adaptar un LLM preentrenado para tareas o dominios específicos mediante un entrenamiento adicional sobre conjuntos de datos más pequeños y relevantes, dentro de sus objetivos se incluyen mejorar la precisión en tareas específicas, infundir conocimiento del dominio y alinear el comportamiento del modelo con las preferencias o instrucciones humanas.

1) *Ajuste Fino Completo (Full Fine-Tuning, FFT)*: Este enfoque tradicional implica actualizar todos los parámetros del modelo preentrenado utilizando los nuevos datos específicos de la tarea, sin embargo, también presenta desafíos significativos debido a su alto costo computacional, en donde se requieren enormes recursos computacionales, múltiples GPUs lo cual lo hace un proceso de alta complejidad, además de que tiende a perder el conocimiento general adquirido durante el preentrenamiento al ser ajustado en tareas más específicas. Este fenómeno ocurre porque el entrenamiento en datos nuevos sobrescribe los pesos que codificaban el conocimiento previo, resultando en una pérdida de capacidades generales y un rendimiento menor en tareas originales [12].

2) *Ajuste Fino Eficiente en Parámetros (Parameter-Efficient Fine-Tuning, PEFT)*: PEFT surge como una solución a los desafíos de FFT que implica ajustar solo un pequeño subconjunto de parámetros o añadir un número reducido de parámetros nuevos, manteniendo estática la mayor parte del modelo preentrenado, dentro de los beneficios se incluyen una reducción drástica de los costes computacionales y de almacenamiento, además de tiempos de entrenamiento mucho más rápidos y un menor riesgo de olvido catastrófico, los métodos pueden introducir nuevos parámetros o módulos:

LoRA (Low-Rank Adaptation) es un método PEFT que introduce dos matrices pequeñas de bajo rango (A y B) cuya multiplicación (BA) representa la actualización de una matriz de pesos preentrenada (W_0) en donde la actualización final es $W=W_0+BA$. Es muy eficiente en parámetros y los pesos LoRA pueden fusionarse con los pesos originales (W_0) después del entrenamiento, eliminando la sobrecarga de inferencia [13].

La evolución desde FFT hacia el diverso panorama de técnicas PEFT está impulsada principalmente por las insostenibles demandas computacionales y de recursos del ajuste fino de modelos con miles de millones de parámetros. Los métodos PEFT están diseñados explícitamente para mitigar estos problemas reduciendo drásticamente el número de parámetros entrenables. Dentro de los pasos para realizar un ajuste fino Raschka, nos presenta tres etapas principales, la preparación de los datos, la ejecución del fine tuning o ajuste fino y la evaluación.

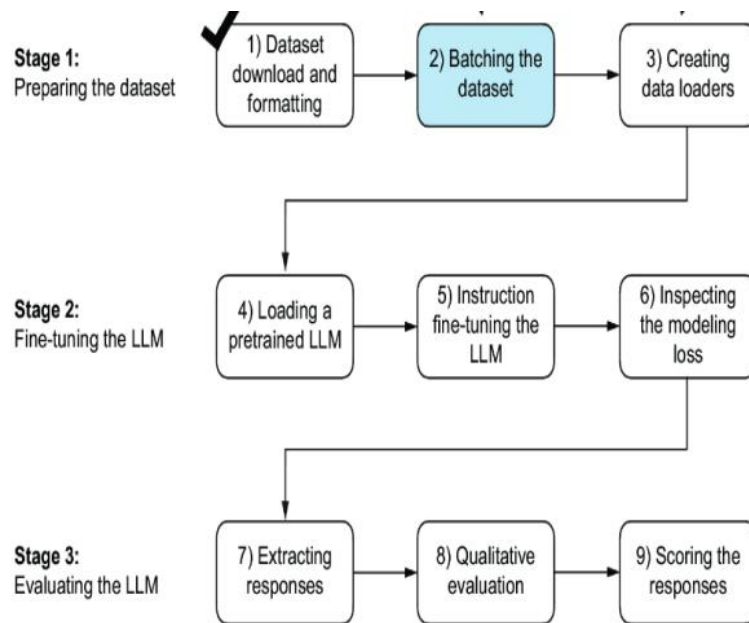


Fig.2. Pasos para realizar un ajuste fino [7]

4.4 Retrieval-Augmented Generation (RAG)

La Generación Aumentada por Recuperación (Retrieval-Augmented Generation, RAG) es una arquitectura híbrida que combina un mecanismo de recuperación de información con un LLM generativo, su justificación radica en abordar las limitaciones inherentes de los LLMs, como la generación de información incorrecta o inventada (alucinaciones), el conocimiento desactualizado debido a la fecha de corte del entrenamiento y la falta de acceso a información específica del dominio o privada, RAG logra esto fundamentando las respuestas del LLM en datos externos y verificables recuperados en tiempo real [14]. Los componentes clave de un sistema RAG son:

1) *Recuperador (Retriever)*: Busca en una base de conocimiento externa la información más relevante basada en la consulta del usuario. Las estrategias de recuperación incluyen búsqueda por palabras clave (BM25), búsqueda semántica basada en vectores (DPR) o enfoques híbridos.

2) *Base de Conocimiento*: La fuente de datos externa cómo documentos, bases de datos o sitios web, que requiere preprocesamiento como la segmentación en fragmentos (chunking) y la creación de embeddings vectoriales.

3) *Generador*: El LLM que sintetiza la información recuperada junto con la consulta original para producir la respuesta final [15].

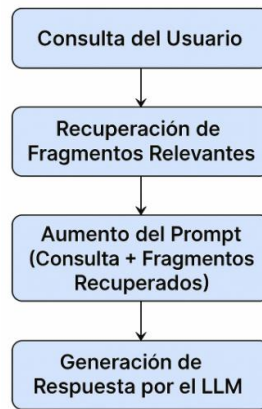


Fig.3. Flujo de RAG. Elaboración propia

A diferencia del ajuste fino, RAG modifica principalmente la entrada al LLM en tiempo de inferencia añadiendo contexto externo, en lugar de alterar los parámetros internos del modelo, RAG es la combinación de un recuperador y un generador, en donde el recuperador obtiene información externa que luego se introduce en el prompt del LLM, el ajuste fino, por el contrario, implica explícitamente ajustar los parámetros del modelo, pero depende en gran medida de la calidad de la recuperación[16].

4.5 Etiquetado de datos

El etiquetado de datos es el proceso de asignar una o varias etiquetas a elementos individuales de un conjunto de datos para que se puedan utilizar en la construcción de modelos de aprendizaje supervisado, este proceso implica la anotación de datos con información relevante que guiará al modelo en su proceso de aprendizaje y toma de decisiones, el etiquetado de texto Incluye tareas como la clasificación de documentos, la detección de sentimientos y el reconocimiento de entidades nombradas (NER), en las que se etiquetan palabras o frases específicas en un texto [7]. El proceso de etiquetado incluye especificar el objetivo del modelo y definir las etiquetas necesarias, recopilar y preparar los datos necesarios para el etiquetado, aplicar las etiquetas manualmente o mediante técnicas automáticas, revisar y validar las etiquetas mediante procesos de revisión y validación.

1) Técnicas de Etiquetado Manual: El etiquetado manual implica que personas revisen y etiqueten cada dato individualmente, es el método más preciso, también es el más intensivo en tiempo y recursos, para ello los anotadores etiquetan los datos basados en procedimientos detallados, posteriormente se comparan y discute para establecer un consenso sobre la etiqueta correcta, también se pueden usar métodos como revisión por pares para evaluar y corregir el etiquetado.

2) Técnicas de Etiquetado Automático: El etiquetado automático utiliza algoritmos y modelos preentrenados para asignar etiquetas a grandes conjuntos de datos sin intervención humana directa, se pueden usar reglas basadas en heurística para etiquetar datos, modelos de clasificación supervisada a través de modelos de aprendizaje automático para predecir etiquetas de nuevos datos, o el reconocimiento de entidades nombradas NER, que utiliza algoritmos para identificar y etiquetar automáticamente.

3) Técnicas de Etiquetado Semiautomático: El etiquetado semiautomático combina la eficiencia del etiquetado automático con la precisión de la revisión humana, en este enfoque, los datos se etiquetan automáticamente y luego se revisan y corrigen por humanos para asegurar la calidad y exactitud, técnicas

como el active learning en donde el modelo identifica los ejemplos más complejos y solicita la intervención humana, mejorando el modelo de forma iterativa son ejemplo de ello.

El etiquetado de datos no es simplemente un prerrequisito, sino una parte integral e iterativa del ciclo de vida de adaptación y evaluación de LLMs, en el ajuste fino se requiere explícitamente datos etiquetados (pares pregunta/respuesta) para adaptar el , [18], la evaluación, especialmente la humana o la comparación con estándares depende de datos etiquetados con precisión para medir el rendimiento [19], además, los resultados de la evaluación a menudo requieren la generación de nuevos datos etiquetados específicos para un mayor ajuste fino lo cual crea un ciclo continuo de etiquetado [20].

Estudios como LIMA [21] sugieren que una cantidad menor de datos de alta calidad curados por humanos puede ser más efectiva para el ajuste fino que grandes cantidades de datos generados automáticamente potencialmente de menor calidad, [22], esto crea una paradoja donde la solución al cuello de botella del etiquetado introduce nuevos problemas de control de calidad, necesitando supervisión humana [13].

4.6 Métricas de evaluación para modelos de lenguaje grande

La evaluación es un paso crítico en el ciclo de vida de los LLMs para comprender la calidad, consistencia, precisión y relevancia de sus resultados, así como su impacto general en la experiencia del usuario, los objetivos de la evaluación incluyen asegurar la fiabilidad y seguridad, prevenir regresiones en el rendimiento tras modificaciones, rastrear mejoras a lo largo del desarrollo y guiar el proceso de ajuste fino.

Evaluar modelos de lenguaje representa un reto debido a múltiples factores, en primer lugar, existe una alta subjetividad, ya que lo que se considera una buena respuesta varía según el contexto y la percepción del evaluador, además, no hay una métrica única que pueda capturar todos los aspectos deseables de la generación de lenguaje, como la coherencia, relevancia, precisión de la respuesta, los errores sutiles y las alucinaciones son otro obstáculo, los LLMs pueden generar respuestas que aparentan ser correctas, pero incluyen errores factuales difíciles de detectar automáticamente, además, la escalabilidad de la evaluación bajo condiciones de uso real o a gran escala presenta una complejidad añadida [23], traducir los valores humanos, que son diversos, complejos y a veces contradictorios, en métricas objetivas y evaluables representa uno de los desafíos más profundos para lograr una verdadera alineación de estos modelos con las expectativas humanas [24], [25]. Estas son algunas métricas utilizadas comúnmente:

4.6.1 Métricas automáticas

BLEU (Bilingual Evaluation Understudy): que compara la similitud entre un texto generado por máquina y uno o más textos de referencia de alta calidad, centrándose principalmente en la fluidez del texto generado mediante la coincidencia de n-gramas o secuencias contiguas de n palabras. Su cálculo se realiza a partir de la precisión de los n-gramas (típicamente de 1 a 4 grados) comparando cuántos n-gramas del texto generado aparecen en los textos de referencia, Incluye una penalización por brevedad para evitar que las traducciones demasiado cortas obtengan puntuaciones altas artificialmente.

Dentro de sus ventajas, se considera una métrica estándar en la traducción automática, computacionalmente eficiente y rápida de calcular por otro lado dentro de sus limitaciones se identifica que no capta bien la similitud semántica, lo que significa que penaliza el uso de sinónimos o paráfrasis aunque el significado sea el mismo lo que puede no correlacionarse bien con el juicio humano sobre la calidad general de la traducción, especialmente en cuanto a adecuación y significado.

ROUGE (Recall-Oriented Understudy for Gisting Evaluation): por su parte es un conjunto de métricas que mide la superposición de n-gramas, secuencias de palabras o pares de palabras entre un resumen generado y uno o más resúmenes de referencia, enfocándose en el recall del contenido de referencia. Su cálculo compara n-gramas (ROUGE-N), secuencias de palabras más largas ROUGE-L, basado en la subsecuencia común más larga, o pares de palabras ROUGE-S entre el texto generado y las referencias, se utiliza la puntuación F1 de precisión y recall de ROUGE para equilibrar la relevancia del contenido generado y la cobertura del contenido de referencia.

Dentro de sus ventajas se evidencia que es una métrica popular para la evaluación de tareas de resumen, es rápida de calcular y se centra en asegurar que el contenido importante de las referencias esté presente en el texto generado en cuanto a sus limitaciones se tiene que al igual que BLEU, puede no capturar completamente el significado semántico y penaliza el uso de sinónimos o estructuras sintácticas ligeramente diferentes, su correlación con el juicio humano sobre la calidad del resumen puede ser limitada [26], [27].

BERTScore (Bidirectional Encoder Representations from Transformers Score): Utiliza embeddings contextuales para calcular la similitud semántica entre los tokens de la salida candidata y la referencia mediante la similitud coseno. Su calculo se realiza a partir de la generación de embeddings para calcular la matriz de similitud coseno y obtener las máximas similitudes por token, calculando BERT Precisión, BERT Recall y BERT F1Score. Dentro de sus ventajas se tiene que captura similitud semántica, maneja mejor paráfrasis sinónimos que métricas basadas en n-gramas, dentro de sus limitaciones se identifica que depende de un texto de referencia y no captura todos los aspectos de calidad.

TABLA 1.
COMPARATIVO DE MÉTODOS DE EVALUACIÓN AUTOMÁTICA

Característica	BLEU	ROUGE	BERTScore
Mecanismo Principal	Compara la precisión de n-gramas con textos de referencia, penalizando la brevedad.	Mide el recall (recuperación) de n-gramas respecto a textos de referencia.	Utiliza embeddings contextuales (de modelos como BERT) para calcular la similitud semántica entre tokens mediante coseno.
Enfoque Primario	Fluidez y precisión de la traducción.	Cobertura del contenido del texto de referencia en el texto generado.	Similitud semántica y significado contextual.
Cuando es mejor su uso	Traducción automática: Cuando se necesita evaluar la fluidez y la coincidencia literal de frases con traducciones de ref.	Evaluación de resúmenes: Cuando el objetivo es asegurar que el resumen cubra puntos clave presentes en resúmenes de ref.	Tareas donde la similitud semántica es crucial: traducción automática, paráfrasis, generación de texto y resúmenes.

4.6.2 Evaluación Humana

La evaluación humana sirve para captar matices, contexto y aspectos cualitativos subjetivos que las métricas automatizadas a menudo pasan por alto, los métodos incluyen juicios binarios, escalas Likert, rankings, clasificaciones categóricas y retroalimentación abierta, dentro de sus principales limitaciones se encuentran el coste, el tiempo, la escalabilidad, la subjetividad inherente a los humanos [27].

Las métricas automatizadas como BLEU, ROUGE o BERTScore son escalables y baratas, pero carecen de matices mientras la evaluación humana capta los matices, pero es cara, lenta y difícil de escalar, por tanto no existe una solución única perfecta, pero la combinación de algunos de estos métodos basados en las necesidades y recursos disponibles, pueden generar un panorama de evaluación más completo que permita tomar decisiones frente a los resultados del entrenamiento de los modelos.

4.7 Antecedentes

A continuación, se exponen tres de las investigaciones revisadas, dos relacionadas con el uso de modelos para mejorar la comunicación entre ciudadanos y entidades del gobierno a través de inteligencia artificial y una correspondiente a la generación de un modelo para consultas médicas a partir de un LLM, a continuación, se describen las principales conclusiones de los estudios, su enfoque metodológico, así como los aportes y diferencias frente al proyecto propuesto.

1) Modelo de atención al público usando el LLM BERT: Basados en BERT, un modelo de lenguaje grande, el equipo de Waston Chua en singapur, en el año 2020 creó a "Super Jamie" un chatbot cuyo objetivo es ser un canal universal de comunicación entre el ciudadano y todas las entidades del gobierno nacional, muchos ciudadanos de Singapur desconocen a qué entidades elevar sus solicitudes o consultas por ende el principal objetivo del proyecto consistió en clasificar el texto que registran los ciudadanos en un agente virtual que sirva para categorizar y direccionar las solicitudes a las agencias del gobierno correspondientes. Partiendo de que muchos ciudadanos han venido realizando preguntas a chatbots dedicados a cada entidad de forma exitosa desde 2014 y de que la mayoría de estas solicitudes son realizadas a las agencias correctas, se usó esta base de datos para direccionar las nuevas solicitudes a través de un único canal de chat [10].

El equipo de Waston Chua partió de la limpieza y organización de los datos suministrados por las diferentes agencias del gobierno, posteriormente usaron BERT, un modelo entrenado y de código abierto para realizar la clasificación y etiquetado de los datos, una vez realizada esta tarea iniciaron con el entrenamiento del modelo, que si bien se encuentra entrenado con bases de datos de corporaciones como Wikipedia, no tenía el acceso a algunos términos usados en el contexto del gobierno de Singapur, el equipo enfrentó esta dificultad reentrenando al modelo con los datos con los que contaban, dividiendo el dataset en 80 % entrenamiento y 20 % validación logrando una exactitud del 85 % en la categorización y direccionamiento. El modelo desarrollado para el gobierno de Singapur es una muestra de cómo se pueden generar nuevas soluciones para contextos específicos a partir de modelos base de código abierto como BERT, sin embargo, se deben tener en cuenta las posibles dificultades que acarrea su adaptación diferentes entornos de entrenamiento, a diferencia de este proyecto, en el Distrito de Santiago de Cali no se cuenta con información previa recolectada a través de los años de la interacción con Chatbots, sino que esta se encuentra en diferentes fuentes de información, en sistemas de gestión documental y repositorios que caracterizan los trámites.

2) *Como la IA generativa puede ser usada efectivamente en el uso de chatbots para el gobierno:* Esta investigación corresponde a un análisis comparativo de diversos chatbots usados en el gobierno en donde se identifica que estos canales se están integrando cada vez más en los servicios gubernamentales para proporcionar respuestas rápidas y personalizadas, reduciendo significativamente la carga de trabajo del personal de servicios públicos, ejemplos notables incluyen "Dave" en el Departamento de Seguridad Económica de Arizona y "Kamu" en Finlandia [11].

La investigación realiza un análisis comparativo entre los modelos ChatGPT y Wenxin Ernie mediante preguntas administrativas variadas en donde se emplean técnicas de análisis de texto, incluyendo análisis de similitud, análisis de frecuencia de palabras y análisis de vectores de palabras, para evaluar las respuestas generadas por ambos modelos, los resultados revelan diferencias significativas en la capacidad de los chatbots para manejar preguntas complejas y proporcionar respuestas útiles y precisas.

El estudio identifica varios desafíos en la aplicación de AIGC en chatbots gubernamentales, tales como sesgos algorítmicos, amenazas a la privacidad y la necesidad de una regulación adecuada, para superar estos retos, se sugieren estrategias de optimización específicas que incluyen la mejora de la precisión de las respuestas, el uso de lenguaje positivo y la adecuación a las necesidades emocionales de los ciudadanos.

Finalmente se concluye que los chatbots como ChatGPT y Wenxin Ernie tienen un potencial significativo para mejorar la interacción entre el gobierno y los ciudadanos, sin embargo, es necesario un enfoque continuo en la mejora de la precisión y la adecuación de las respuestas.

3) *Chatbot médico creado a partir del LLM BERT:* La investigación describe el desarrollo y los resultados de un chatbot médico basado en BERT (Bidirectional Encoder Representations from Transformers) para mejorar la comunicación en el sector salud, los autores Arun Babu y Sekhar Babu utilizaron varios conjuntos de datos médicos como MIMIC-III, BioASQ, PubMed y COVID-19. MIMIC-III que incluyen notas clínicas, diagnósticos, medicamentos, resultados de laboratorio y otra información médica, las técnicas usadas para el procesamiento de texto incluyeron tokenización, eliminación de palabras vacías, técnicas de NLP como stemming y lematización, vectorización, n-grams y etiquetado de partes del discurso, se entrenó el LLM BERT en los datos médicos procesados para aprender el contexto y la importancia de las palabras, ajustando sus pesos para mejorar el rendimiento en tareas médicas específicas como la clasificación de textos, el reconocimiento de entidades y la respuesta a consultas.

En las pruebas del modelo identificaron que el chatbot mantiene el contexto de la conversación para seguir las consultas previas del usuario y proporcionar respuestas relevantes y coherentes, el sistema identifica la intención del usuario, como buscar consejo médico, programar una cita o solicitar información sobre una condición específica, y genera respuestas informativas utilizando plantillas basadas en reglas y métodos de recuperación, el chatbot basado en BERT alcanzó una precisión del 98% en la gestión de consultas médicas, con una puntuación de precisión del 97%, una puntuación AUC-ROC del 97%, un recall del 96% y una puntuación F1 del 98%, lo cual demuestra su capacidad para manejar consultas médicas con un alto nivel de precisión y fiabilidad, del estudio se resalta la variedad de las técnicas usadas para el procesamiento de texto que contribuyeron al alto desempeño del modelo [12].

5. ENTENDIMIENTO Y PREPARACIÓN DE LOS DATOS

5.1 Fuentes de datos disponibles

El flujo de aprobación y actualización en el distrito de Santiago de Cali se lleva a cabo a través del sistema de información SGT (Sistema de gestión de trámites) en el cual reposan los datos sobre las hojas de vida de los trámites, en él se realiza el flujo de creación y aprobación de estos de acuerdo con las modificaciones legales que haya lugar o necesidades de la entidad, la Subdirección de Trámites, Servicios y Gestión Documental es la responsable de mantenerlo actualizado y solicitar la publicación en el sistema de información SUIIT (Sistema Único de Información de Trámites) fuente única y válida de la información de los trámites que todas las instituciones del estado ofrecen a la ciudadanía y que se puede consultar de forma pública.

Al identificar las fuentes de información se encuentran aspectos a considerar para lograr determinar cuál es la fuente más propicia para generar los datos etiquetados requeridos para el entrenamiento del modelo, por un lado SUIIT es la página oficial de la nación, sin embargo el acceso a la base de datos es restringida debido a que no corresponde a un sistema de información propio como sí lo es el SGT, por otro lado en el SUIIT es donde se encuentra la información aprobada por el DAFP y que se considera es válida para publicación al ciudadano bajo los requisitos de exigidos por la nación, sin embargo antes de pasar al sistema de información de la nación todos los trámites son aprobados inicialmente a través del sistema propio SGT. Esta es la razón principal para que el sistema de información seleccionado para obtener los datos sea el Sistema de Gestión de Trámites, de donde se generó un reporte que permitiera caracterizar los datos disponibles, de aquí se logró identificar lo siguiente:

El reporte se generó de una base de datos tipo relacional o SQL de donde se exportó un archivo en formato CSV el cual cuenta con las siguientes tablas:

TABLA 2.
INFO GENERAL DESCRIPCIÓN

Nombre de Columna (Variable)	Tipo de Variable	Escala de Medición	Descripción de la Variable	Ejemplos de Categorías / Rango de Valores
ID_TRAMITE	Cuantitativa	Nominal	Identificador único numérico para cada trámite o servicio.	Ej: 101, 102, ... , 350 (Depende del rango real)
CODIGOSUIT	Cualitativa	Nominal	Código identificador del trámite según el Sistema Único de Información de Trámites (SUIT).	Ej: 9885, 12345, AB123 (Depende del formato real)
NOMBRE_TRAMITE	Cualitativa	Nominal	Nombre oficial completo del trámite o servicio.	Ej: "Asignación de Nomenclatura", "Impuesto Predial Unificado"
DESCRIPCION_TRAMITE	Cualitativa	Nominal	Resumen o descripción breve de en qué consiste el trámite.	Ej: "Obtener la dirección oficial de un predio.", "Pago del impuesto..."
ORGANISMO_RESPONSABLE	Cualitativa	Nominal	Departamento, secretaría o entidad de la Alcaldía encargada del trámite.	Ej: "Departamento Administrativo de Planeación Municipal", "Secretaría de Hacienda"
TIEMPO_RESPUESTA_ESTIMADO	Cualitativa	Nominal	Plazo oficial o estimado para la resolución o respuesta del trámite.	Ej: "15 días hábiles", "Inmediato", "30 días"
TIENE_COSTO	Cualitativa	Nominal (Binaria)	Indica si el trámite o servicio tiene un costo asociado (Sí/No).	Sí, No
MODALIDAD	Cualitativa	Nominal	Forma en que se puede realizar el trámite.	Ej: Presencial, En línea, Mixto
LINK_INFORMACION	Cualitativa	Nominal	URL a la página web con información detallada del trámite.	Ej: https://www.cali.gov.co/tramite/... ,

Trámites InfoGeneral: Esta tabla presenta la Información general sobre trámites, el identificador de cada trámite, descripción, tipo y enlaces relacionados con el trámite.

TABLA 3.
COSTOS

Nombre de Columna (Variable)	Tipo de Variable	Escala de Medición	Descripción de la Variable	Ejemplos de Categorías / Rango de Valores
ID_TRAMITE (o CODIGOSUIT)	Cualitativa	Nominal	Identificador único del trámite.	Ej: 101, 9885, AB123
TIPO_CONDICION	Cualitativa	Nominal	Categoría de la condición (ej. Requisito, Documento, Excepción).	Ej: "Requisito general", "Documento obligatorio", "Excepción aplica si..."
DESCRIPCION_CONDICION	Cualitativa	Nominal	Detalle de la condición o requisito específico que debe cumplir el ciudadano.	Ej: "Ser mayor de edad", "Estar al día en impuestos prediales"
DOCUMENTO_SOPORTE	Cualitativa	Nominal	Documento que se debe presentar para validar la condición (si aplica).	Ej: "Cédula de ciudadanía", "Certificado de paz y salvo"
OBSERVACIONES	Cualitativa	Nominal	Notas adicionales o aclaraciones sobre la condición.	Ej: "Aplica para personas naturales y jurídicas"

Costos: En esta hoja se detallan los costos relacionados con cada trámite, en donde si el trámite tiene costo, especifica cada uno de los costos asociados al mismo.

TABLA 4.
VERIFICACIONES

Nombre de Columna (Variable)	Tipo de Variable	Escala de Medición	Descripción de la Variable	Ejemplos de Categorías / Rango de Valores
ID_TRAMITE (o CODIGOSUIT)	Cuantitativa/Cualitativa	Nominal	Identificador único del trámite.	Ej: 101, 9885, AB123
TIPO_CONDICION	Cualitativa	Nominal	Categoría de la condición (ej. Requisito, Documento, Excepción).	Ej: "Requisito general", "Documento obligatorio", "Excepción aplica si..."
DESCRIPCION_CONDICION	Cualitativa	Nominal	Detalle de la condición o requisito específico que debe cumplir el ciudadano.	Ej: "Ser mayor de edad", "Estar al día en impuestos prediales"
DOCUMENTO_SOPORTE	Cualitativa	Nominal	Documento que se debe presentar para validar la condición (si aplica).	Ej: "Cédula de ciudadanía", "Certificado de paz y salvo"
OBSERVACIONES	Cualitativa	Nominal	Notas adicionales o aclaraciones sobre la condición.	Ej: "Aplica para personas naturales y jurídicas"

Verificaciones: Teniendo en cuenta que los trámites cuentan con requisitos para poderse realizar, en esta tabla se detallan cada uno de los requisitos o verificaciones requeridas.

TABLA 5
PASOS PARA REALIZAR TRÁMITES

Nombre de Columna (Variable)	Tipo de Variable	Escala de Medición	Descripción de la Variable	Ejemplos de Categorías / Rango de Valores
ID_TRAMITE (o CODIGOSUIT)	Cuantitativa/Cualitativa	Nominal	Identificador único del trámite.	Ej: 101, 9885, AB123
NUMERO_PASO	Cuantitativa	Ordinal	Número secuencial que indica el orden del paso dentro del trámite.	Ej: 1, 2, 3, ...
DESCRIPCION_PASO	Cualitativa	Nominal	Descripción detallada de la acción a realizar en este paso.	Ej: "Radicar documentos en ventanilla", "Realizar pago en línea"
LUGAR_REALIZACION_PASO	Cualitativa	Nominal	Ubicación física o virtual donde se debe realizar el paso.	Ej: "Ventanilla Única de Atención", "Página web de la Alcaldía"
REQUISITOS_PASO	Cualitativa	Nominal	Documentos o condiciones específicas necesarias para este paso.	Ej: "Copia de cédula", "Formulario diligenciado"
DURACION_ESTIMADA_PASO	Cualitativa	Nominal	Tiempo aproximado que toma completar este paso específico.	Ej: "15 minutos", "2 horas", "Inmediato"

Pasos Trámites: Además de los requisitos, los trámites cuentan con una serie de pasos necesarios para realizarse, en esta tabla se indica paso por paso, cuáles son los requeridos para realizar el trámite.

TABLA 6
ESTAMPILLAS

Nombre de Columna (Variable)	Tipo de Variable	Escala de Medición	Descripción de la Variable	Ejemplos de Categorías / Rango de Valores
ID_TRAMITE (o CODIGOSUIT)	Cuantitativa/Cualitativa	Nominal	Identificador único del trámite al que se asocia la estampilla.	Ej: 101, 9885, AB123
NOMBRE_ESTAMPILLA	Cualitativa	Nominal	Nombre oficial de la estampilla requerida.	Ej: "Estampilla Pro-Cultura", "Estampilla Pro-Desarrollo Urbano"
VALOR_ESTAMPILLA	Cuantitativa	Razón	Costo o valor de la estampilla.	Ej: 5000, 12000.50, "Según UVT"
BASE_GRAVABLE	Cualitativa/Cuantitativa	Nominal/Razón	Base sobre la cual se calcula el valor de la estampilla (si aplica).	Ej: "Valor del contrato", "Monto del avalúo"
NORMATIVA_ESTAMPILLA	Cualitativa	Nominal	Referencia a la ordenanza, acuerdo o ley que rige la estampilla.	Ej: "Acuerdo Municipal 0123/2024", "Ley XYZ de 2023"
OBSERVACIONES	Cualitativa	Nominal	Notas adicionales o condiciones específicas sobre la estampilla.	Ej: "Aplica solo para contratos superiores a X SMLMV"

Estampilla: Si el trámite requiere de estampillas, en esta tabla se encuentran los datos relacionados con las estampillas requeridas.

5.2 Descripción de los datos del reporte de información general de trámites

En el conjunto de datos analizado, se identificaron un total de 245 trámites, dentro de los cuales 219 de los trámites cuentan con enlace a una hoja de vida documentada, mientras que 26 trámites no disponen de esta información que garantiza que el trámite fue aprobado y se encuentra publicado en el sistema de información de la nación SUIT, a su vez al revisar si se cuenta con la descripción de los pasos requeridos para realizar el trámite en la tabla “pasos” se identificó que los mismos datos no cuentan con la especificación de los pasos, información que se considera fundamental para el propósito del proyecto.

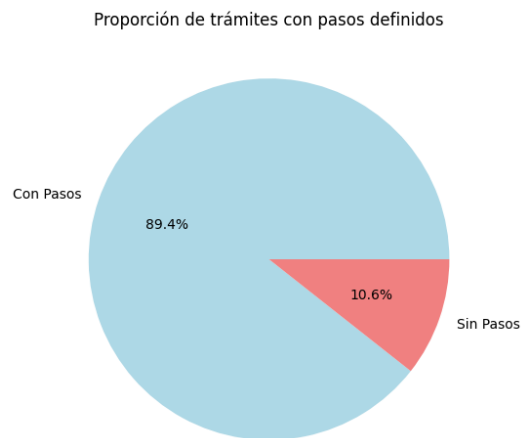


Fig.4. Porcentaje de trámites sin pasos



Fig.5. Porcentaje de trámites sin hoja de vida

Respecto a la disponibilidad de un enlace para realizar el trámite en línea, 64 trámites cuentan con un link habilitado, mientras que 181 trámites no tienen esta opción, lo que implica que requieren un proceso presencial. Los trámites están categorizados como “Presencial”, lo cual indica que el ciudadano debe desplazarse hasta una oficina para llevarlo a cabo, “Parcialmente en línea”, que significa que parte del procedimiento es virtual y parte presencial, y “en línea”, lo que indica que el trámite se puede realizar en su totalidad de forma remota.

Distribución de Estado de Automatización

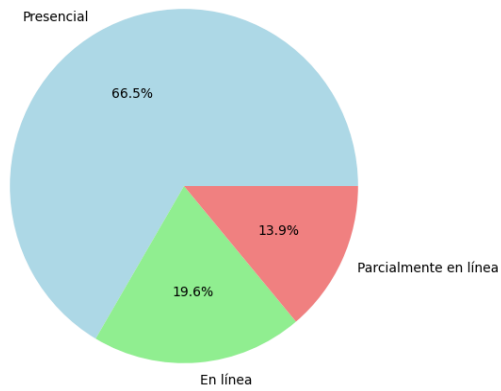


Fig.6. Porcentaje de trámites en línea y con enlace disponible

Proporción de trámites con enlace para gestión en línea

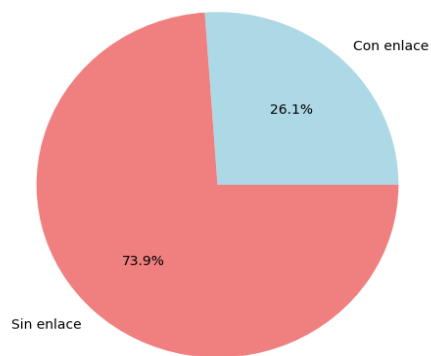


Fig. 7. Porcentaje de trámites con enlace disponible

Por otro lado, se identificó que 102 trámites tienen un costo asociado, mientras que 143 trámites son gratuitos, esto implica que más de la mitad de los trámites pueden realizarse sin un pago obligatorio, lo cual es un dato relevante para los usuarios que buscan acceder a los trámites sin incurrir en gastos adicionales.

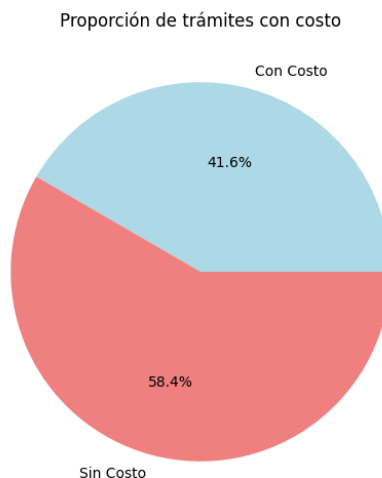


Fig.8. Porcentaje de trámites con costo asociado

Del reporte de trámites del sistema de información SGT y a partir del análisis descriptivo realizado se logró identificar que algunos de los trámites no cuentan con la información completa para ser tenidos en cuenta en el entrenamiento del modelo en donde 219 de los 245 trámites del reporte se encuentran documentados por completo y existen 26 registros con datos faltantes, de los cuales se identificó posteriormente que corresponden a trámites que están en proceso de actualización y de los cuales no se cuenta con la información completa, por tanto y teniendo en cuenta que se requiere de la información completa y actualizada para lograr generar las preguntas y respuestas requeridas para entrenar el modelo, se optó por imputar estos registros eliminándolos del reporte suministrado.

Posteriormente y a partir del reporte depurado se elaboraron preguntas para los 219 trámites asociadas a la información disponible a partir de los datos del reporte, además fueron elaboradas teniendo en cuenta la información suministrada por el equipo de ventanilla única quienes indicaron el tipo de orientaciones más recurrentes que solicitan los ciudadanos, en donde se abordaron temas como la definición general del trámite, su duración, costo, los pasos requeridos para realizarlo, el organismo responsable de atender dicho trámite, así como en donde se puede realizar.

Las preguntas y respuestas para entrenar los modelos fueron creadas de forma manual en un archivo tipo xlsx a partir de las variables del reporte imputado en donde se tuvieron en cuenta las siguientes variables:

- Nombre del trámite
- Descripción
- Enlace a la hoja de vida
- Tiempo de respuesta
- Pasos a realizar
- Verificación nombre

Para el dataset original se elaboraron 1380 preguntas y respuestas, sin embargo una vez se realizaron los primeros entrenamientos de los modelos seleccionados, se identificó la necesidad de ampliar el dataset

hasta las 3500 preguntas y respuestas para lograr que el modelo generalizara correctamente. A continuación, se muestra un ejemplo de preguntas y respuestas elaboradas.

TABLA 7.
EJEMPLO DE PREGUNTAS Y RESPUESTAS

Pregunta	Respuesta deseada
¿En qué consiste el trámite asignación de nomenclatura?	El trámite Asignación de Nomenclatura consiste en: Obtener el número de identificación único de los bienes raíces que se encuentren dentro de la jurisdicción.
¿Cuál es el tiempo de respuesta del trámite asignación de nomenclatura?	El trámite Asignación de Nomenclatura tiene un tiempo de La resolución de Asignación de Nomenclatura tiene un tiempo de 15 Día(s) hábil(es).
¿El trámite Asignación de Nomenclatura tiene algún costo?	El trámite Asignación de Nomenclatura no tiene costo.
¿Cómo se puede realizar el trámite Asignación de Nomenclatura?	Para realizar el trámite Asignación de Nomenclatura, sigue los siguientes pasos: 1. Reunir documentos y cumplir condiciones: Formulario Solicitud Asignación de Nomenclatura disponible en https://www.gov.co/ficha-tramites-y-servicios/T11481 2.Radicar documentos
¿Qué organismo es responsable del trámite Asignación de Nomenclatura?	El organismo responsable del trámite Asignación de Nomenclatura es el Departamento Administrativo de Planeación Municipal.
¿Dónde puedo realizar el trámite Asignación de Nomenclatura?	Consulta los puntos de atención en la hoja de vida del trámite: https://www.gov.co/ficha-tramites-y-servicios/T11481

Fuente: Elaboración propia

5.3 Preprocesamiento de los datos

El objetivo primordial de la etapa de preprocesamiento fue adecuar el conjunto de datos crudo con las preguntas y respuestas elaboradas para el entrenamiento y lograr comparar el desempeño de los modelos seleccionados en la tarea específica para la orientación en trámites, esta fase fue fundamental, ya que la calidad y el formato de los datos de entrada impactó directamente en el rendimiento y la capacidad de generalización de los modelos entrenados.

La base de datos para ambos entrenamientos se originó a partir de un conjunto de datos sobre trámites, alojado en Hugging Face Hub, para el modelo Llama 3.2 3B se utilizó el dataset referenciado como giosan8918/tramites9, mientras que para el modelo Phi 3.5 mini se empleó una versión denominada giosan8918/tramites10,ambos con la misma cantidad de datos, diferenciados únicamente por el formato aplicado para el cargue, esto dado que el uso de Hugging Face Hub requiere un formato específico como

repositorio, el cargue en este ambiente facilitó el acceso y la carga recurrente de datos en los respectivos entornos de Google Colab, donde se ejecutaron los diferentes procesos de entrenamiento.

Posteriormente, se procedió con la división del conjunto de datos utilizando la función `train_test_split` de la librería `datasets`, el dataset se segmentó en tres conjuntos distintos: un 80% destinado al entrenamiento del modelo (conjunto de entrenamiento), un 10% para la validación durante el proceso de ajuste fino (conjunto de validación), y el 10% restante reservado para la evaluación final del rendimiento del modelo (conjunto de prueba). El formateo del dataset se aplicó a todas las particiones entrenamiento, validación y prueba, mediante la función `standardize_sharegpt` de la librería `Unsloth`.

Una vez estandarizado el formato y dividido el dataset, se aplicó la plantilla de chat específica para cada modelo a sus respectivas conversaciones, para el primer modelo, `unsloth/Llama-3.2-3B-Instruct`, se empleó la función `get_chat_template` de `Unsloth` configurada para `llama-3.1`, dicha plantilla estructura cada intercambio conversacional mediante identificadores especiales como:

```
<|begin_of_text|>,<|start_header_id|>, <|end_header_id|> y <|eot_id|>
```

De igual forma se aplicó para el segundo modelo, `unsloth/Phi-3.5-mini-instruct`, pero configurada con la plantilla `phi-3`, que utiliza delimitadores como:

```
<|user|> y <|assistant|> seguidos de <|end|>
```

En ambos casos, se definió una función específica, `formatting_prompts_func`, que utilizó `tokenizer.apply_chat_template` para generar el texto formateado, en esta etapa, la tokenización (`tokenize=False`) se difirió para ser manejada directamente por el `SFTTrainer`. Ejemplo de formato aplicado:

```
<|user|> Explícame el trámite Determinantes para el ajuste de un Plan Parcial?<|end|>  
<|assistant|> Este trámite tiene como propósito fijar las delimitaciones y la información sobre los determinantes aplicables para el ajuste o modificación de un plan parcial aprobado.<|end|>
```

Con los datos ya formateados específicamente para cada modelo, se realizaron los preparativos para su utilización con el `SFTTrainer` de la librería `TRL`, para el agrupamiento y la preparación eficiente de los lotes de datos durante el entrenamiento, se empleó `DataCollatorForSeq2Seq` en el caso del modelo `Llama-3.1`, con el fin de optimizar los tiempos de preprocesamiento, se configuró el procesamiento paralelo para operaciones de mapeo y tokenización.

Una vez contamos con los datos formateados y tokenizados se procedió con el entrenamiento de los modelos, se debe tener en cuenta que los LLMs modernos son entrenados sobre vastos corpus de texto natural, lo que les permite aprender intrínsecamente las funciones contextuales de las stopwords, la relevancia estructural de la puntuación y las relaciones entre diferentes formas léxicas de las palabras por tanto no se aplicó un flujo de preprocesamiento tradicional en cuanto la eliminación de estos elementos, que podría de hecho omitir información contextual valiosa y afectar negativamente su capacidad para comprender y generar texto de manera natural y coherente, especialmente en un dominio como es el de trámites, en donde la precisión del lenguaje y la estructura de las preguntas y respuestas puede agregar información importante a los modelos.

6. SELECCIÓN DE MODELOS DE LENGUAJE GRANDE

6.1 Definición de criterios de selección

Para lograr establecer cuáles modelos de lenguaje grande son adecuados para realizar el Fine-Tuning primero se partió de la identificación de los modelos disponibles que contarán al menos con las siguientes características:

1) *Entrenado en español*: El modelo debe haber sido entrenado o tener capacidades avanzadas en español, esto es fundamental ya que el modelo se utilizará eventualmente para interactuar con personas que hablan y escriben en español, lo que garantiza que las respuestas sean precisas, coherentes y relevantes, dado que la mayoría de las interacciones serán llevadas a cabo en este idioma.

2) *Posibilidad de realizar Fine-Tuning*: El modelo debe ser flexible para realizar el fine-tuning con datos específicos de la Alcaldía de Cali, lo que permite personalizar las respuestas basadas en las hojas de vida de los trámites disponibles.

3) *Código abierto*: Cualquier equipo de desarrollo puede descargar el modelo, estudiarlo y adaptarlo a sus necesidades sin tener que pagar por licencias para ajustar los parámetros del modelo de tal forma que responda mejor a tareas o contextos específicos, como en el caso de los trámites sin incurrir en costos adicionales.

4) *Visión del uso de los LLMs*: Para obtener una visión objetiva de los LLMs de código abierto disponibles que cumplen con los criterios fundamentales, se consultó la plataforma Artificial Analysis (artificialanalysis.ai). Esta plataforma ofrece comparaciones y benchmarks independientes basados en investigaciones sobre modelos de inteligencia artificial, se recopiló datos de varios modelos destacados, cuyas características y métricas de rendimiento se resumen en la Tabla 8 que cuenta con los siguientes elementos:

a) *Nombre*: Es el nombre del modelo de lenguaje. Los modelos tienen diferentes versiones y capacidades, y suelen ir acompañados de detalles como la cantidad de parámetros que contienen, lo que influye en su capacidad de procesamiento y generación de lenguaje.

b) *Creador*: Indica la empresa o el equipo que desarrolló el modelo, las principales empresas que desarrollan estos modelos suelen ser grandes compañías tecnológicas como Meta, Google, Mistral, o OpenAI, equipos de investigación académica o colaboraciones de código abierto.

c) *Ventana de Contexto*: Se refiere a la cantidad de tokens que el modelo puede manejar en una sola interacción, cuanto mayor sea la ventana de contexto, más información puede recordar el modelo en una sola consulta o conversación.

d) *Índice (Promedio Normalizado)*: Es una métrica que representa el rendimiento global del modelo en una variedad de tareas, se normaliza para facilitar la comparación entre diferentes modelos, cuanto mayor es el valor, mejor es el rendimiento del modelo en comparación con otros.

e) *Arena de Chatbots*: Este valor mide el rendimiento del modelo en pruebas de chatbot, donde se evalúa la capacidad de respuesta del modelo en entornos de conversación, un valor más alto indica un mejor desempeño en la generación de diálogos coherentes y útiles.

f) MMLU (Massive Multitask Language Understanding): es una métrica que evalúa la capacidad del modelo para manejar y responder preguntas sobre una amplia gama de temas, desde matemáticas hasta ciencias sociales.

TABLA 8.
COMPARATIVO ENTRE MODELOS
GRANDES DE LENGUAJE DE CÓDIGO ABIERTO

Modelo	Creador	Ventana de Contexto	Índice (Promedio Normalizado)	Arena de Chatbots	MMLU
Llama 3.1 405B	Meta	128k	72	1266	0.886
Llama 3.2 90B (Visión)	Meta	128k	66	1249	0.860
Llama 3.2 11B (Visión)	Meta	128k	54	1172	0.694
Llama 3.1 8B	Meta	128k	53	1172	0.694
Llama 3.2 3B	Meta	128k	47	1148	0.777
Phi 3.5 Mini instruct	Microsoft	4K	22	1037	0.435
Gemma 2 27B	Google	8k	61	1212	0.752
Gemma 2 9B	Google	8k	46	1185	0.713
Mistral Large 2	Mistral	128k	73	1251	0.84
Mixtral 8x22B	Mistral	65k	61	1148	0.777
Pixtral 12B	Mistral	128k	56	N.D	N.D

Fuente: <https://artificialanalysis.ai/leaderboards/models>

6.2 Aplicación de criterios de selección

Para determinar cuáles modelos de lenguaje grande podría ser entrenados con datos del Distrito de Santiago de Cali para dar respuesta a consultas sobre trámites, fue necesario conocer cuáles son las necesidades y recursos disponibles, para ello se llevaron a cabo sesiones con el equipo del Departamento Administrativo de Tecnologías de la Información y las Comunicaciones para establecer cuáles eran las necesidades que debería abordar el problema en una fase inicial, que permitiera determinar la relevancia, veracidad y completitud de las respuestas obtenidas y continuar con una posible segunda fase de implementación, de aquí se definieron los siguientes criterios específicos para la selección de los modelos a entrenar:

1) Abarcar la mayor cantidad de población al menor costo (Costo): En una ciudad con una diversidad de población urbana y rural como Santiago de Cali, es fundamental que el sistema sea accesible a todos los ciudadanos, independientemente de su ubicación geográfica o nivel de acceso a tecnología, esto implica la necesidad de implementar una solución que funcione de manera eficiente en entornos como la página web de la Alcaldía, aplicaciones móviles y plataformas de mensajería.

El costo del servicio es un elemento prioritario para tener en cuenta dado que se espera que se le pueda dar continuidad a través de los proyectos de inversión o de recursos de funcionamiento que maneja el Distrito, para ello la viabilidad financiera es un aspecto fundamental, en donde históricamente se ha observado que proyectos de transformación digital a través de diferentes aplicativos, que requieren de grandes recursos para su funcionamiento se quedan sin disponibilidad por falta de recursos, por tanto la solución debe ser financieramente viable para, esto implica considerar el tamaño del modelo, los requisitos de hardware para inferencia y fine-tuning, y los costos operativos asociados, favoreciendo modelos que ofrezcan un buen equilibrio entre rendimiento y costo.

2) Una solución escalable (Escalabilidad): La Alcaldía de Cali enfrenta una demanda creciente de solicitudes sobre trámites, servicios y eventos. Inicialmente, el enfoque debe estar en consultas de trámites, pero la solución debe ser capaz de crecer y abarcar otros aspectos como servicios y eventos, además, preferiblemente permitiendo funciones adicionales como el reconocimiento de imágenes o voz especialmente útiles para ciudadanos que puedan tener barreras de acceso a la tecnología o dificultades en la interacción escrita. Al ser escalable, el modelo garantizará que la plataforma no se quede obsoleta a medida que las necesidades y demandas evolucionen, permitiendo una respuesta adaptada y eficiente en cualquier ámbito solicitado.

3) Menor intervención humana posible (autonomía): La implementación de una solución basada en un modelo de lenguaje grande (LLM) debe estar diseñada para minimizar la intervención humana en las consultas frecuentes, mejorando la eficiencia del servicio y reduciendo los costos operativos., dado que el equipo de atención al ciudadano tiene una alta demanda de solicitudes que no se procesan dentro de los tiempos estipulados y que el personal de la entidad presenta alta rotación, además de que gran parte de los recursos se gestionan a través de proyectos de inversión los cuales pueden o no tener continuidad dependiendo de los cambios y decisiones tomadas en los gobiernos de turno, una solución con menor intervención humana implica posibilidad de mayor continuidad en el tiempo.

Que sea factual en cuanto a las respuestas, con una variabilidad reducida en la capacidad generativa, dado que la correcta clasificación y el manejo adecuado de los trámites es clave para mejorar la eficiencia

del servicio, es fundamental que el modelo de lenguaje sea capaz de proporcionar respuestas precisas, las respuestas deben ser claras, consistentes y basadas en información actualizada para evitar errores o confusión, una alta variabilidad en las respuestas generadas por el modelo podría generar desinformación, afectando la confianza en el servicio y, por ende, la reputación de la entidad, sin embargo no se requiere que el modelo sirva únicamente un sistema de recuperación de información, la capacidad generativa debe ser adaptable y brindar respuestas acertadas de acuerdo a las consultas que se realicen en lenguaje natural.

4) Seguridad suficiente para evitar brechas que permitan la manipulación de respuestas (seguridad): La seguridad es un pilar fundamental en la implementación de cualquier solución digital, especialmente en el ámbito gubernamental, las solicitudes relacionadas con trámites frecuentemente implican el manejo de datos personales y sensibles, lo que aumenta el riesgo de explotación de vulnerabilidades si no se implementan mecanismos de seguridad adecuados para prevenir fraudes, suplantación de identidad y manipulación de información, el modelo podría fracasar, por tanto es necesario que permita la incorporación de medidas de protección, garantizando así la confidencialidad y la integridad de las respuestas y los datos proporcionados.

1) Costo: El costo relativo se evaluó considerando el tamaño del modelo y los recursos necesarios para su implementación y operación, modelos más grandes, como LLama 3.1 405B, requieren un poder de cómputo considerable, lo que implica costos elevados en hardware y consumo energético, especialmente si se requiere reentrenamiento o ajustes frecuentes, lo que contrasta con modelos más compactos como LLaMA 3.2 3B, LLaMA 3.2 1B y Phi-3.5 Mini Instruct los cuales ofrecen un desempeño aceptable con una significativa reducción en costos operativos.

2) Escalabilidad: La escalabilidad está relacionada con la capacidad del modelo para adaptarse a un aumento en la demanda o en el volumen de datos. Modelos como Mixtral 8x22B y Llama 3.1 405B poseen una alta capacidad para procesar múltiples consultas simultáneas y manejar grandes contextos. Sin embargo, modelos más pequeños como LLaMA 3.2 3B y Phi-3.5 Mini Instruct tienen una escalabilidad más limitada.

3) Seguridad: Los modelos de mayor capacidad, como Llama 3.1 405B y Gemma 2 27B, permiten configuraciones más sofisticadas de seguridad y control de salidas, incluyendo el ajuste fino de respuestas y aplicación de filtros contextuales. Por otro lado, modelos como LLaMA 3.2 1B y Phi-3.5 Mini Instruct, si bien no ofrecen todas estas funcionalidades avanzadas, pueden mantener niveles aceptables de seguridad cuando se combinan con mecanismos externos de validación y supervisión, especialmente en aplicaciones donde el riesgo es bajo o está controlado.

4) Autonomía: La autonomía refiere a la capacidad del modelo para mantener la coherencia y precisión de sus respuestas sin necesidad de intervención constante. Modelos como Gemma 2 27B y Mistral Large 2 destacan en este aspecto gracias a su amplio conocimiento y estabilidad conversacional. En el caso de modelos pequeños, como Llama 3.2 3B y Phi-3.5 Mini Instruct presentan un desempeño sorprendentemente robusto en tareas de tipo instrucción o guías paso a paso, por lo que puede operar de manera autónoma en flujos bien definidos, siempre que se limite a dominios específicos como la orientación institucional.

5) Alcance: El alcance se refiere a la diversidad de funciones que puede desempeñar el modelo, incluyendo capacidades multimodales. Modelos como LLama 3.2 90B (Visión) y Pixtral 12B, que integran capacidades

visuales, tienen un mayor alcance en entornos complejos. frente a modelos más pequeños como Phi-3.5 Mini tienen un alcance más limitado al manejo de texto, pero son altamente eficientes en contextos específicos donde no se requiere procesamiento de imágenes, como es el caso de sistemas de orientación sobre trámites, en los que la claridad y precisión del lenguaje natural es prioritaria.

Tras la identificación de modelos y la definición de los criterios fundamentales y específicos del proyecto discutidos previamente, se procedió a un análisis ponderado para la preselección de los LLMs más adecuados para la fase piloto. Este análisis consideró el énfasis del proyecto en el costo-eficiencia y la viabilidad de implementación en un entorno con potenciales restricciones presupuestarias, sin sacrificar la capacidad de cumplir con los objetivos funcionales básicos de orientación sobre trámites. Para formalizar este proceso, se asignaron ponderaciones a los criterios contextualizados del proyecto, reflejando su importancia relativa para la Alcaldía de Cali en esta etapa inicial:

- Costo y Eficiencia de Recursos: Ponderación Alta (40%) - Crítico para la sostenibilidad.
- Autonomía: Ponderación Alta (30%) - Esencial para la confianza y utilidad del servicio.
- Alcance Funcional Inicial (Texto, Trámites): Ponderación Media (15%) - El enfoque principal.
- Seguridad: Ponderación Media (10%) - Importante, pero con mitigaciones externas posibles.
- Escalabilidad: Ponderación Baja (5%) - Consideración para fases posteriores, no prioritaria para la viabilidad de un piloto inicial.

A continuación, se aplicaron los criterios a los modelos preseleccionados de la Tabla 1 con un enfoque en aquellos que ya mostraban un buen balance inicial de tamaño y capacidad, aunque una cuantificación más precisa requeriría de benchmarks extensos y específicos del dominio en todos los modelos, se realizó una estimación cualitativa basada en la información disponible como el tamaño del modelo, métricas MMLU/Arena.

Las puntuaciones de 1 a 5 son asignaciones conceptuales para ilustrar el proceso de decisión, donde 5 es el más favorable para ese criterio específico, la puntuación ponderada estimada es el resultado de multiplicar la puntuación de cada criterio por su ponderación y sumar los resultados.

TABLA 9.
CRITERIOS DE SELECCIÓN APLICADOS A
MODELOS PRESELECCIONADOS

Criterio (Ponderación)	Llama 3.1 405B (Meta)	Mistral Large 2 (Mistral)	Gemma 2 27B (Google)	Llama 3.1 8B (Meta)	Llama 3.2 3B (Meta)	Phi 3.5 Mini Instruct (Microsoft)
Costo (40%)	Muy Bajo (1/5)	Bajo (2/5)	Medio-Bajo (2.5/5)	Alto (4/5)	Alto (4/5)	Muy Alto (5/5)
Autonomía (30%)	Muy Alto (5/5)	Muy Alto (5/5)	Alto (4/5)	Medio-Alto (3.5/5)	Medio-Alto (3.5/5)	Medio (3/5)
Alcance (15%)	Muy Alto (5/5)	Muy Alto (5/5)	Alto (4/5)	Alto (4/5)	Alto (4/5)	Alto (4/5)
Seguridad (Intrínseca del Modelo) (10%)	Alto (4/5)	Alto (4/5)	Alto (4/5)	Medio (3/5)	Medio (3/5)	Medio (3/5)
Escalabilidad (5%)	Muy Alto (5/5)	Muy Alto (5/5)	Alto (4/5)	Medio (3/5)	Medio (3/5)	Bajo-Medio (2.5/5)
Puntuación	2.95	3.35	3.325	3.675	3.725	3.925

De la calificación anterior se tiene que, si bien los modelos de mayor escala como Llama 3.1 405B y Mistral Large 2 se destacan por su autonomía, consistencia y alcance, su baja puntuación en costo, el criterio de mayor ponderación para este proyecto los hace menos viables para una implementación inicial bajo las restricciones presupuestarias y de infraestructura disponibles.

Los modelos Llama 3.2 3B y especialmente Phi 3.5 Mini Instruct emergen como los candidatos más fuertes al considerar el balance ponderado de todos los criterios, estos modelos destacan por su excelente costo eficiencia en relación con su tamaño manejable el cual reduce significativamente los requisitos computacionales tanto para el proceso de ajuste fino como para la inferencia en producción, alineándose con la necesidad de una solución sostenible, en donde también cuenta con métricas que si bien no alcanzan los niveles de los modelos más grandes, ofrecen un buen rendimiento en relación con su tamaño, indicando una buena base de conocimiento y capacidad, ofreciendo potencial para la tarea requerida, especialmente tras un ajuste fino con datos del dominio de trámites. Por lo tanto, con base en este análisis ponderado y la argumentación previa, los modelos base seleccionados para el ajuste fino en este proyecto aplicado fueron:

- Llama 3.2 3B, un modelo de Meta que ofrece un excelente equilibrio entre tamaño, capacidad para seguir instrucciones y eficiencia de recursos.

- Phi 3.5 Mini, un modelo de Microsoft, altamente eficiente y compacto, ideal para entornos con recursos limitados sin sacrificar completamente el rendimiento en tareas conversacionales e instructivas.

7. MODELADO CON MODELOS DE LENGUAJE GRANDE

7.1 Entrenamiento de modelos seleccionados

Para el entrenamiento se utilizaron los modelos unsloth/Llama-3.2-3B-Instruct y unsloth/Phi-3.5-mini-instruct que corresponden a modelos optimizados que incluyen la posibilidad de carga en 4 y 16 bits lo cual reduce significativamente el consumo de memoria haciendo posible realizar el fine-tuning en GPUs con VRAM limitada como la Tesla T4 proporcionada por el entorno de Google Colab.

Se aplicó un fine-tuning supervisado para enseñar al modelo a generar respuestas a partir de ejemplos de preguntas y sus correspondientes respuestas deseadas, se utilizó LoRA (Low-Rank Adaptation) para adaptar el modelo a la nueva tarea modificando solo un pequeño porcentaje de los parámetros totales lo cual aceleró el entrenamiento y redujo los requisitos de memoria comparado con un ajuste fino completo y teniendo en cuenta los recursos disponibles, Inicialmente se cargaron los modelos pre entrenados con cuantización de 4 bits para reducir el uso de memoria, y con una longitud máxima de secuencia (`max_seq_length`) de 2048.

La fase inicial de experimentación se centró en evaluar la capacidad de generalización de los LLMs seleccionados utilizando el conjunto de datos original compuesto por 1380 pares de preguntas y respuestas, durante esta etapa exploratoria, se llevaron a cabo múltiples ciclos de entrenamiento de los cuales se presentan ejemplos en el Anexo C, variando sistemáticamente los hiperparámetros clave como la tasa de aprendizaje (`learning_rate`), el tamaño del lote (`per_device_train_batch_size` junto con `gradient_accumulation_steps`), el número de épocas de entrenamiento (`num_train_epochs`) y los parámetros del optimizador (`optim`, `weight_decay`), también se experimentó con el parámetro `temperature` evaluando valores muy cercanos a 0, e incluso 0 absoluto, con el objetivo de minimizar la aleatoriedad en la generación y promover respuestas más factuales, adicionalmente, se exploraron valores de `temperature` moderados para encontrar un equilibrio que permitiera respuestas naturales y coherentes sin introducir excesiva creatividad o alucinaciones que pudieran derivar en información incorrecta, esta exploración se complementó con ajustes en otros parámetros de muestreo como `top_k` (limitando la selección de tokens a los 'k' más probables) y `min_p` (o `top_p / nucleus sampling`, que selecciona el conjunto más pequeño de tokens cuya probabilidad acumulada excede un umbral `p`), buscando optimizar la calidad, precisión y consistencia de las respuestas generadas.

A pesar de estos esfuerzos en el ajuste de hiperparámetros, se observó que ninguno de los modelos lograba una capacidad de generalización satisfactoria, tendiendo a alucinar en la mayoría de las respuestas, además de no responder de la forma en que dichas respuestas deseadas estaban estructuradas en los datos de entrada. Este fenómeno, comúnmente asociado con el `underfitting` o un aprendizaje insuficiente debido a la escasez o baja diversidad de los datos de entrenamiento, se manifestaba en respuestas poco precisas o incompletas al ser evaluados con ejemplos no vistos durante el entrenamiento, posteriormente se identificó que la limitada cantidad de datos iniciales restringía la habilidad de los modelos para capturar la variabilidad del dataset de entrada. Ante esta limitación, y reconociendo la importancia fundamental del volumen y la calidad de los datos en el rendimiento de los LLMs, se procedió a una fase de aumento de datos, técnica que no solo implicó incrementar el número de datos, sino también enriquecer la diversidad y complejidad de las formulaciones de las preguntas y la estructura de las respuestas, manteniendo siempre la fidelidad semántica con la información original de los trámites, de esta forma el conjunto de datos se

expandió a 3500 entradas como se indicó en el capítulo 5.2 Preprocesamiento de datos . Una vez se realizó este cambio en los datos de entrada se logró observar una mejora notoria en la capacidad para generalizar de los modelos, en donde la combinación de hiparámetros que surgió mejor efecto para cada uno de los modelos entrenados se presenta a continuación.

TABLA 10
HIPERARÁMETROS DEFINIDOS PARA MODELOS
CON MEJOR DESEMPEÑO

Hiperparámetro de SFTTrainer	Modelo 1: Llama 3.2	Modelo 2: Phi 3.5	Descripción
dataset_num_proc	2	2	Número de procesos utilizados para el preprocesamiento de datos (tokenización, mapeo de funciones). Valor > 1 acelera esta etapa.
gradient_accumulation_steps	8	4	Número de pasos durante los cuales se acumulan gradientes antes de una actualización de pesos. Permite simular un tamaño de lote mayor
warmup_steps	100	100	Número de pasos iniciales donde la tasa de aprendizaje aumenta linealmente desde 0 hasta learning_rate. Ayuda a estabilizar el entrenamiento temprano.
num_train_epochs	25	10	Número total de veces que el conjunto de datos de entrenamiento completo se presenta al modelo.
learning_rate	5,00E-06	5,00E-06	Tasa de aprendizaje inicial para el optimizador
logging_steps	1	1	Frecuencia (en pasos) con la que se registran las métricas de entrenamiento.
optim	"adamw_8bit"	"adamw_8bit"	Optimizador AdamW con estados cuantizados a 8 bits, reduciendo el uso de memoria.
weight_decay	1	1	Coefficiente de regularización L2 para penalizar pesos grandes y prevenir el sobreajuste.

El modelo Llama 3.2 3B Instruct tuvo un tiempo de entrenamiento total de 4.48 horas, mientras el modelo Phi 3.5 Mini Instruct completó su entrenamiento en 3.95 horas, utilizando la GPU Tesla T4

disponible en el entorno de Google Colab. Aunque el modelo Llama 3.2 3B fue entrenado durante más épocas (25 frente a 10 para Phi), el modelo Phi 3.5 Mini tuvo un mayor número total de pasos de actualización de gradientes (1750 frente a 1075 para Llama) debido a su menor tamaño de lote efectivo (16 frente a 64 para Llama). La técnica LoRA permitió ajustar solo una pequeña fracción de los parámetros totales para ambos modelos 1.30% para Llama y 1.47% para Phi.

La gráfica de pérdida (training loss) y validación de entrenamiento para el modelo Llama 3.2 3B, entrenado durante 25 épocas (1075 pasos), mostró una tendencia general descendente tanto en la línea de pérdida como de validación.

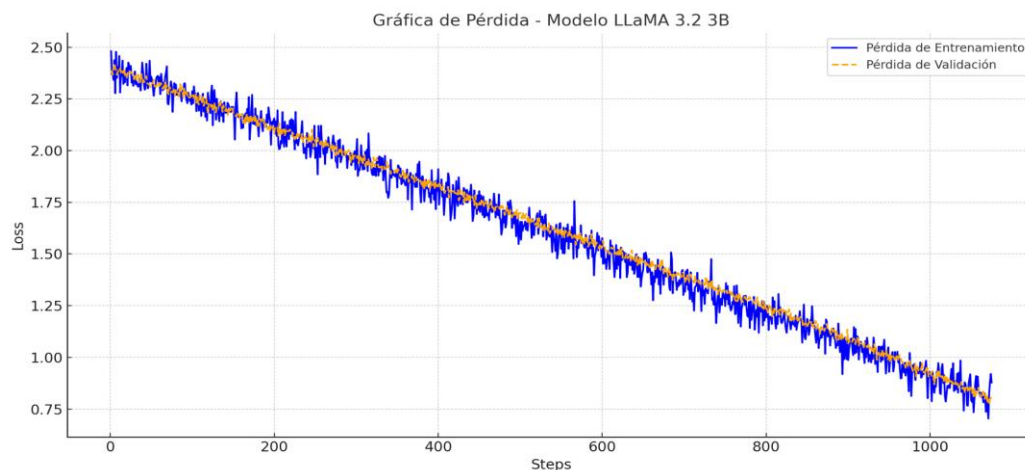


Fig.9. Gráfica de pérdida para modelo Llama 3.2 3B

La disminución constante y la eventual estabilización cerca de indican que el modelo converge hacia una configuración de pesos que minimiza el error en los datos de entrenamiento, una disminución sostenida en la pérdida de entrenamiento es una señal de que el modelo está aprendiendo y ajustando sus parámetros internos para minimizar la función de error definida sobre el conjunto de entrenamiento, esto demuestra que el modelo tiene la capacidad suficiente para capturar los patrones presentes en los datos de entrenamiento, el hecho de que la pérdida de validación también disminuya inicialmente indica que el aprendizaje del modelo sobre los datos de entrenamiento se está generalizando correctamente sobre datos no vistos.

Para el modelo Phi 3.5 mini, entrenado durante 10 épocas y 1750 pasos, se observan resultados similares en donde desciende rápidamente al inicio y se estabiliza con pequeñas fluctuaciones hacia el final.

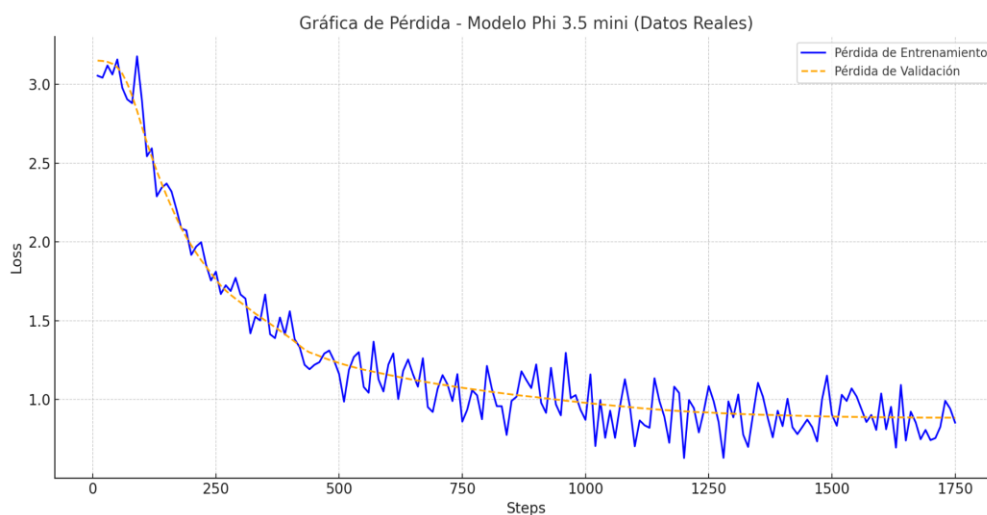


Fig. 10. Gráfica de pérdida para modelo Phi 3.5

Ambas gráficas tanto para el modelo LLama 3.2 3B como para el modelo Phi 3.5 mini indican que los modelos lograron aprender sobre los datos proporcionados sin tener sobre ajuste, ubicándose en el rango de pérdida entre 0.5 y 1, alrededor de 0.8 lo cual indica un aprendizaje equilibrado del conjunto de datos proporcionado, los modelos aprendieron progresivamente, las dos curvas convergen hacia valores cercanos sin una brecha creciente.

7.2 Evaluación de modelos entrenados

7.2.1 Generación de respuestas a partir de los modelos entrenados

Antes de la evaluación de desempeño cuantitativa, el primer paso de esta etapa fue realizar pruebas en el modelo generando las entradas o prompts para validar que las respuestas generadas fueran coherentes y tuvieran relación con los datos de entrenamiento, se probó con cincuenta preguntas generadas automáticamente, similares a las utilizadas para entrenar el modelo, de tal forma que la respuesta permitiera dar un concepto sobre el entrenamiento realizado. Ejemplos Modelo Llama-3.2-3B-Instruct :

a) *Prompt*: ¿Cuánto tiempo tarda el trámite Asignación de Nomenclatura? Respuesta Obtenida: La cantidad de tiempo que tarda el trámite Asignación de Nomenclatura es de 15 Día(s) hábil(es).

b) *Prompt*: ¿Es gratuito el trámite Registro de establecimientos veterinarios?" Respuesta Obtenida: Este trámite tiene un costo

c) *Prompt* : Cual es la definición del trámite: Modificación de área y linderos por acuerdo mutuo? Respuesta Obtenida: Es el trámite que se realiza cuando los propietarios de predios colindantes deciden modificar el lindero común, basándose en un acuerdo mutuo. Este proceso requiere un levantamiento topográfico para reflejar la nueva configuración de los linderos.

2) Ejemplos Modelo Phi-3.5-mini-instruct

a) *Prompt*: ¿Cuánto tiempo tarda el trámite Asignación de Nomenclatura? Respuesta Obtenida: El trámite Asignación de Nomenclatura tiene un tiempo de respuesta estimado de 15 Día(s) habil(es).

b) *Prompt* : Explícame el trámite Determinantes para el ajuste de un Plan Parcial. Respuesta Obtenida: Este trámite tiene como propósito fijar las delimitaciones y la información sobre los determinantes aplicables para el ajuste o modificación de un plan parcial aprobado.

c) *Prompt* : ¿Qué institución maneja el trámite Autorización para movimiento de tierras? Respuesta Obtenida: El trámite Autorización para movimiento de tierras es manejado por el Departamento Administrativo de Planeación

De la evaluación cualitativa de respuestas se logró identificar que los modelos efectivamente habían logrado aprender de los datos proporcionados, generalizando correctamente la estructura de las respuestas, en términos generales se apreciaban respuestas coherentes, sin embargo en cuanto a la factualidad o precisión de las mismas se encontró que existían debilidades marcadas para los tipos de respuestas que se relacionaban con los pasos requeridos para realizar los trámites, así como aquellas que indican los lugares donde realizar los trámites, casos en los que ambos modelos tendían a alucinar.

7.2.2 Evaluación cuantitativa con BERTSCORE

Tras la fase de evaluación inicial a partir de las estadísticas de entrenamiento y generación de respuestas, se procedió a realizar una evaluación cuantitativa del rendimiento de los modelos, para esta evaluación, se empleó la métrica BERTScore, la cual mide la similitud semántica entre las respuestas generadas por el modelo y las respuestas de referencia del conjunto de prueba. BERTScore calcula la precisión o cuán relevantes son los tokens generados, el recall o cuántos tokens relevantes de la referencia fueron capturados y una puntuación F1 la media armónica de precisión y recall, ofreciendo una medida robusta de la calidad y coherencia de las respuestas. Los resultados obtenidos para cada modelo en el conjunto de prueba fueron los siguientes:

Modelo Llama-3.2-3B-Instruct

- Precisión BERTScore: 0.7077
- Recall BERTScore: 0.8211
- F1-Score BERTScore: 0.7583

Modelo unsloth/Phi-3.5-mini-instruct

- Precisión BERTScore: 0.7834
- Recall BERTScore: 0.8527
- F1-Score BERTScore: 0.8144

Los puntajes obtenidos indican que ambos modelos, tras el ajuste fino, lograron un grado considerable de alineación semántica con las respuestas esperadas, el modelo Phi-3.5-mini-instruct presentó un desempeño superior en todas las métricas BERTScore en comparación con el modelo Llama-3.2-3B-Instruct y un F1-Score en el rango de 0.81 lo que indica que el modelo Phi-3.5 Mini fue capaz de generar respuestas que no solo eran contextualmente relevantes sino que también cubrían una proporción significativa de la información contenida en las respuestas de referencia.

Durante la evaluación cualitativa, que implicó la inspección manual de las respuestas generadas, se observó que ambos modelos eran capaces de generar respuestas coherentes y, para ciertos tipos de preguntas, factuales y completas de acuerdo con la información del dataset de entrenamiento, sin

embargo, a pesar de estos resultados especialmente para el modelo Phi-3.5 Mini, la evaluación cualitativa reveló una limitación persistente: la incidencia de alucinaciones o la generación de información no factual, estas alucinaciones no pudieron ser completamente mitigadas mediante la manipulación de los parámetros de generación como la temperature reduciéndola a valores cercanos a 0 para minimizar la aleatoriedad o los parámetros de muestreo como top_k y si bien estos ajustes ayudaron a mejorar la factualidad hasta cierto punto, no eliminaron por completo la generación de respuestas incorrectas de acuerdo con el corpus de conocimiento específico de los trámites con el que fueron entrenados.

La presencia residual de alucinaciones se consideró un factor decisivo ya que para una aplicación en el sector público que brindara información oficial a ciudadanos, la factualidad y la fiabilidad de las respuestas son de suma importancia y un sistema que, aunque generalmente coherente, presenta instancias de información incorrecta, no alcanzaría el nivel de desempeño y confianza esperado para su implementación operativa, por consiguiente, y a pesar de los avances logrados mediante el ajuste fino supervisado de estos LLMs, se determinó que el nivel de control sobre la factualidad de las respuestas, incluso con la optimización de los parámetros de generación, no era suficiente para los estándares requeridos por el proyecto.

8. MODELADO CON RAG

8.1 Aplicación de Retrieval-Augmented Generation (RAG)

La Generación Aumentada por Recuperación (RAG) fue seleccionada debido a su capacidad para tomar las respuestas de un modelo de lenguaje grande (LLM) a partir de un corpus de conocimiento específico, lo cual reduce significativamente la generación de contenido impreciso o no factual. Para probar esta alternativa se desarrolló un pipeline RAG utilizando principalmente la librería LangChain, con los siguientes componentes:

1) *Preparación y procesamiento de datos:* Se utilizó la data de preguntas y respuestas base para entrenar los modelos previos y se cargó a través de un documento accesible públicamente, se utilizó `WebBaseLoader` de `langchain_community.document_loaders` para cargar el contenido del documento publicado que contiene todas las preguntas y respuestas usadas para entrenar los modelos previos.

El texto extraído se segmentó en fragmentos (chunks) utilizando `RecursiveCharacterTextSplitter.from_tiktoken_encoder`, se definió un tamaño de fragmento de 250 caracteres y un solapamiento de 0 caracteres buscando crear fragmentos concisos para la recuperación. Para la conversión de los fragmentos de texto y las consultas de los usuarios en representaciones vectoriales o embeddings, se empleó `OpenAIEmbeddings` de `langchain_openai` y se utilizó una clave API de OpenAI directamente en el código para este propósito.

2) *Cargue de datos en una base de datos vectorial:* Los embeddings generados para los fragmentos del documento se almacenaron y se consultaron utilizando `SKLearnVectorStore` de `langchain_community.vectorstores` que es una base de datos vectorial, el recuperador se configuró para obtener los $k=4$ fragmentos más relevantes (basados en similitud de coseno) para cada pregunta.

El componente encargado de generar la respuesta final fue un modelo Llama 3.2 servido localmente a través de Ollama. específicamente, se instanció `ChatOllama` de `langchain_ollama` con `model="llama3.2"` y una `temperature=0` para promover respuestas más precisas o factuales y abordar la problemática que se tenía con los modelos a los cuales se les realizó el fine tuning.

LangChain se utilizó para construir el pipeline RAG, definiendo una cadena que primero recupera el contexto relevante y luego lo pasa, junto con la pregunta, al LLM llama3.2 en la generación de respuestas basadas en el contexto recuperado y las directrices del rol de asistente:

3) *Plantilla utilizada:*

Eres un asistente experto en trámites públicos del Distrito de Santiago de Cali.
Usa los documentos proporcionados para responder de forma clara y precisa.
Si no encuentras la información, indica que no está disponible.

El flujo operativo del sistema RAG implementado fue el siguiente:

- Una pregunta del usuario es recibida por la `RAGApplication`.

- La pregunta es utilizada por el retriever (basado en SKLearnVectorStore y OpenAIEmbeddings) para buscar y recuperar los 4 fragmentos de texto más relevantes del corpus documental.
- Los contenidos de estos fragmentos recuperados se concatenan para formar el contexto.
- Este contexto y la pregunta original se insertan en la plantilla de prompt.
- El prompt completo y aumentado se envía al LLM ChatOllama (modelo llama3.2 3B).
- El LLM genera una respuesta basada en el contexto y la pregunta, adhiriéndose a las instrucciones de la plantilla.
- La respuesta generada se devuelve como salida.

Para evaluar la efectividad del sistema RAG, se utilizó un conjunto de datos de preguntas y respuestas de referencia cargado desde el archivo Excel "Tramites_Preguntas_Respuestas_ajustado manualmente.xlsx". Se generaron respuestas para estas preguntas utilizando la RAG Application y se compararon con las respuestas de referencia mediante varias métricas.

4) Ejemplo de respuestas Modelo RAG Llama 3.1 3B

a). *Prompt:* ¿Hay algún trámite que me sirva para corregir linderos? Respuesta Obtenida: El trámite Rectificación de área y linderos es un procedimiento que tiene como objetivo obtener el registro de la corrección por rectificación de áreas y linderos en las bases del catastro.

Para completar este trámite, se deben seguir los siguientes pasos:

- Reunir los documentos: Cédula de ciudadanía con fotocopia(s), Escritura pública donde aparezca el área a corregir con fotocopia(s), Poder con original(es) y Plano general de localización del predio con cotas amarradas a las coordenadas de Cali con original(es).
- Realizar el pago.
- Radicar documentación requerida.
- Atender Visita Técnica en el predio para demostrar el área del predio como resultado de la inspección catastral.

El trámite tiene un costo asociado y el tiempo de respuesta estimado es de 30 días hábiles. Es importante destacar que este trámite no es gratuito, ya que requiere un pago.

8.2 Resultados de la evaluación

Al observar los resultados obtenidos con el modelo RAG se confirmó que efectivamente este ayuda notablemente a reducir las alucinaciones que presentaban modelos previos, sobre todo cuando se trata de respuestas relacionadas con los pasos o el tiempo requerido para realizar los trámites, lo cual hace que este modelo sea más confiable para su aplicación que los modelos a los cuales se les realizó el ajuste fino.

Resultados de la evaluación cuantitativa.

- Precisión BERTScore: 0.8167
- Recall BERTScore: 0.8993
- F1-Score BERTScore: 0.8551

Los resultados en terminos generales de BERTScore a su vez presentan una mejora notable comparados con las evaluaciones previas a los modelos a los cuales se les realizó el ajuste fino.

8.3 Evaluación comparativa

8.3.1 Evaluación cuantitativa consolidada

A lo largo de este proyecto, se implementaron y evaluaron tres enfoques principales para la tarea de orientación sobre trámites:

- Ajuste Fino Supervisado (SFT) del modelo unsloth/Llama-3.2-3B-Instruct.
- Ajuste Fino Supervisado (SFT) del modelo unsloth/Phi-3.5-mini-instruct.
- Un sistema de Generación Aumentada por Recuperación (RAG) utilizando OpenAIEmbeddings para la recuperación y un modelo llama3.2 (servido vía Ollama) para la generación.

La evaluación cuantitativa se centró principalmente en la métrica BERTScore (Precisión, Recall y F1-Score) para medir la similitud semántica entre las respuestas generadas y un conjunto de respuestas de referencia, para el sistema RAG. La siguiente tabla resume los resultados de BERTScore obtenidos para cada uno de los enfoques evaluados, utilizando la misma cantidad de datos para el conjunto test de referencia:

TABLA 11.
RESULTADOS DE MODELOS UTILIZADOS

Métrica BERTScore	Modelo Llama 3.2 3B Instruct (SFT)	Modelo Phi 3.5 Mini Instruct (SFT)	Sistema RAG (Llama 3.2 3B)
Precisión BERTScore	0.70	0.78	0.81
Recall BERTScore	0.82	0.85	0.89
F1-Score BERTScore	0.75	0.81	0.85

El sistema RAG claramente mostró el mejor desempeño cuantitativo general, alcanzando un F1-Score de BERTScore de 0.8551, este valor es significativamente más alto que los obtenidos por ambos modelos SFT, evaluados, el Recall BERTScore de 0.8993 para el sistema RAG es particularmente destacable, indicando que las respuestas generadas por RAG capturaron una proporción muy alta de la información contenida en las respuestas de referencia, lo cual es importante para un sistema de orientación sobre trámites ya que que las respuestas son más completas, cómo se observó en el ejemplo previú. La Precisión BERTScore de

0.8167 también es superior a la de los modelos SFT, lo que implica que los tokens generados por el RAG son más pertinentes y menos propensos a incluir información no relevante en comparación con las respuestas de referencia.

8.3.2 Evaluación humana

Para obtener una perspectiva complementaria sobre el desempeño de los modelos, se implementó una fase de evaluación humana con un enfoque de puntuación binaria, esta evaluación fue conducida con la participación de tres profesionales del Departamento Administrativo de Desarrollo e Innovación Digital de la Alcaldía de Santiago de Cali, quienes poseen conocimiento sobre los trámites y los estándares de calidad para la información dirigida a la ciudadanía. A los participantes se les explicó detalladamente el protocolo, la evaluación se centró en calificar las respuestas generadas por los tres sistemas de LLM identificados anónimamente durante la evaluación pero correspondientes a RAG ("Modelo 1"), Llama 3.2 3B SFT ("Modelo 2") y Phi 3.5 Mini SFT ("Modelo 3").

Para cada una de las 10 preguntas de prueba seleccionadas, a los evaluadores se les presentó la pregunta original (prompt de entrada) y una respuesta considerada deseable, de acuerdo a los datos usados para entrenar el modelo, de tal forma que pudieran comparar aspectos como la factualidad de la respuesta.

El Sistema de calificación para cada criterio evaluado en cada respuesta generada, los expertos debían asignar una puntuación binaria:

- 1: Si el criterio se cumplía satisfactoriamente.
- 0: Si el criterio no se cumplía.

Los criterios específicos utilizados para evaluar cada respuesta generada fueron:

- Claridad y estructura: ¿La respuesta es clara y bien estructurada?
- Precisión de la respuesta a la pregunta: ¿La respuesta responde exactamente lo que se pregunta?
- Suficiencia de la Información: ¿La respuesta da información suficiente?
- Corrección y coincidencia con la Información Original (Factualidad): ¿La respuesta es correcta y coincide con la información original?

La puntuación obtenida por cada modelo, para cada pregunta y por cada evaluador, se basó en la suma de los "1s" asignados a través de los cuatro criterios, para determinar un indicador global de rendimiento, se calculó un "porcentaje de cumplimiento" para cada modelo, este porcentaje se obtuvo de la siguiente manera: Para cada evaluador, se sumaron todas las puntuaciones "1" obtenidas por un modelo a través de las 10 preguntas y los 4 criterios, esta suma se dividió entre la puntuación máxima posible que ese modelo podría haber alcanzado con ese evaluador (10 preguntas * 4 criterios * 1 punto = 40 puntos), el resultado se multiplicó por 100 para obtener el porcentaje de cumplimiento de ese evaluador para ese modelo. Finalmente, se promediaron los porcentajes de cumplimiento de los tres evaluadores para cada modelo, obteniendo así una puntuación global consolidada.

Los porcentajes de cumplimiento globales promediados para cada modelo, basados en la evaluación de los tres expertos, fueron los siguientes:

TABLA 12
RESULTADOS DE EVALUACIÓN POR EXPERTOS

Modelo Evaluado	Evaluador 1	Evaluador 2	Evaluador 3	Porcentaje de Cumplimiento Global Promedio (%)
Modelo 1 (RAG)	90,00%	87,50%	90,00%	89,17%
Modelo 2 (Llama SFT)	62,50%	70,00%	72,50%	68,33%
Modelo 3 (Phi SFT)	67,50%	60,00%	70,00%	65,83%

De acuerdo con estos resultados se identificó que el Modelo 1 (RAG) obtuvo el mayor porcentaje de cumplimiento global promedio, alcanzando un 89,17%. Este resultado es significativamente superior al de los otros dos modelos indicando la calidad de sus respuestas a través de los cuatro criterios definidos y las preguntas evaluadas, un porcentaje alto que implica que el Modelo RAG cumplió satisfactoriamente con la mayoría de los aspectos de claridad, precisión, suficiencia en la gran mayoría de las instancias en donde los evaluadores individuales también mostraron una preferencia marcada y consistente por el modelo.

8.4 Análisis de resultados

Como se mostró en la tabla 11, los resultados de BERTScore, que miden la similitud semántica entre las respuestas generadas y las respuestas de referencia, fueron los siguientes:

- Llama SFT: F1-Score = 0.7583
- Phi SFT: F1-Score = 0.8144
- RAG Llama 3.2: F1-Score = 0.8551

Estos resultados indicaron que ambos modelos SFT lograron un nivel considerable de alineación semántica tras el ajuste fino, siendo el Phi SFT ligeramente superior al Llama SFT y que el sistema RAG superó a ambos modelos, obteniendo el F1-Score más alto lo cual indica que las respuestas generadas por RAG no solo son semánticamente más similares a las respuestas ideales, sino que también son más completas, esto se pudo corroborar con la evaluación humana realizada en donde se reveló que el modelo RAG Llama 3.2 fue percibido como significativamente superior, cumpliendo con los criterios de calidad definidos para la evaluación en casi el 90% de los casos por otra parte los modelos a los cuales se les realizó el ajuste fino mostraron un cumplimiento considerablemente menor, rondando el 65-68% aunque fueron capaces de generar respuestas, fallaron con mayor frecuencia en los criterios de factualidad principalmente lo cual impactó considerablemente su desempeño percibido.

Un aspecto para destacar es como él Llama SFT tuvo un ligero margen sobre el Phi SFT, contrario a los resultados obtenidos en BERTScore, lo que muestra cómo los criterios humanos priorizaron

aspectos que no son el foco principal de la similitud semántica de BERTScore para estos dos modelos.

9. CONCLUSIONES Y TRABAJOS FUTUROS

9.1 Conclusiones

El presente proyecto de investigación aplicada realizó la exploración y evaluación de diferentes enfoques basados en Modelos de Lenguaje Grande (LLMs) con el objetivo de entrenar un modelo que fuera capaz de brindar orientación sobre trámites de la Alcaldía de Santiago de Cali, a través de un proceso iterativo que abarcó la preparación de datos, el ajuste fino supervisado (SFT) de modelos de código abierto, y la implementación de una arquitectura de Generación Aumentada por Recuperación (RAG), obteniendo resultados concluyentes que aportan a la construcción de la solución final.

Se demostró la viabilidad técnica de realizar un ajuste fino supervisado de LLMs como Llama 3.2 3B Instruct y Phi 3.5 Mini Instruct en entornos con recursos computacionales limitados (GPU Tesla T4) y gracias a técnicas de optimización como la cuantización de 4 bits y LoRA (Low-Rank Adaptation), ambos modelos mostraron capacidad de aprendizaje y adaptación al dominio específico de los trámites, como lo evidenciaron las curvas de pérdida y validación descendentes durante el entrenamiento así como la capacidad de generar respuestas semánticamente coherentes con los datos suministrados, generalizando correctamente la estructura con la que fueron entrenados.

En la evaluación cuantitativa mediante BERTScore, el modelo Phi 3.5 Mini Instruct (F1-Score: 0.8144) superó ligeramente al Llama 3.2 3B Instruct (F1-Score: 0.7583) en términos de similitud semántica con las respuestas de referencia, sin embargo, la evaluación humana, mostró una limitación crítica en ambos modelos, la persistencia de alucinaciones y la generación de información no factual, incluso después de la optimización de hiperparámetros de generación como las épocas de entrenamiento y la temperatura, lo cual los hace poco fiables para una aplicación de cara al ciudadano donde la veracidad es primordial.

Por otra parte la implementación de un sistema RAG, utilizando embeddings para la recuperación y un modelo llama3.2 para la generación, demostró ser el enfoque más efectivo alcanzando un F1-Score de BERTScore de 0.8551, superando a ambos modelos SFT, como también lo logró en la evaluación humana con un porcentaje de cumplimiento global promedio del 89,17%, significativamente más alto que los modelos SFT. Esta alta puntuación, otorgada por expertos, resalta su superioridad en criterios críticos como la factualidad, claridad, precisión y suficiencia de la información.

El enfoque RAG demostró ser significativamente más efectivo en mitigar las alucinaciones y mejorar la factualidad de las respuestas, debido a que la generación se fundamentó en fragmentos de información recuperados, se obtuvo el mejor rendimiento cuantitativo con un (F1-Score de BERTScore de 0.8551) y fue el preferido de manera contundente en la evaluación humana (89,17% de cumplimiento global promedio).

Se constató la importancia crítica de la evaluación combinada cuantitativa y cualitativa para obtener una comprensión más completa del rendimiento de los modelos, mientras BERTScore ofreció información valiosa sobre la similitud semántica y desempeño estimado de los modelos, la evaluación humana fue indispensable para identificar problemas de factualidad y valorar la utilidad real de las respuestas principalmente en la comparación de resultados entre el modelo Phi que presentaba un mejor resultado cuantitativo frente a Llama, en donde el factor humano logró determinar cuál modelo ofrecía mejores respuestas de acuerdo con la evaluación cualitativa realizada.

En definitiva, si bien el ajuste fino de LLMs como Llama 3.2 3B y Phi 3.5 Mini demostró ser técnicamente factible y capaz de lograr una alineación semántica considerable, la persistencia de alucinaciones limitó su aplicación directa mientras el sistema RAG se presentó como la arquitectura superior, ofreciendo un control mucho mayor sobre la factualidad y logrando un desempeño sobresaliente tanto en métricas automáticas como en la evaluación por expertos del dominio, por tanto este enfoque se perfila como la ruta más clara para desarrollar una herramienta de orientación ciudadana fiable y precisa para la Alcaldía de Santiago de Cali.

9.2 Trabajos futuros

Dentro de los trabajos futuros se considera relevante el diseñar y desarrollar una interfaz de usuario intuitiva como tipo chatbot integrado en la web de la Alcaldía o en canales de mensajería como para interactuar con el sistema RAG, realizando pruebas piloto con un grupo controlado de usuarios tanto ciudadanos y funcionarios para recopilar feedback sobre la usabilidad, utilidad y precisión del sistema, ajustando según se considere. Si los recursos lo permiten, se considera pertinente investigar la integración de capacidades multimodales (procesamiento de imágenes o voz) para mejorar la accesibilidad, como se consideró en los criterios iniciales de escalabilidad.

El expandir el alcance del sistema para cubrir no sólo trámites, sino también servicios, eventos e información general de interés para los ciudadanos de Cali, lo que requerirá la consolidación de la data relacionada con los servicios de la entidad.

También se considera pertinente realizar un análisis detallado del costo total de la solución RAG, considerando tanto los costos de desarrollo, infraestructura local o en la nube, como los posibles costos de APIs y evaluar la sostenibilidad a largo plazo de la solución dentro del presupuesto y la infraestructura tecnológica del Distrito.

10. ANEXOS

Anexo A: Muestra del Conjunto de Datos de Entrenamiento y Evaluación

Este anexo presenta una selección de pares de preguntas y respuestas utilizadas durante las fases de entrenamiento, validación y prueba de los modelos. Estos ejemplos ilustran la diversidad y el formato de los datos empleados, extraídos del archivo "Tramites_Preguntas_Respuestas_ajustado manualmente .csv"

Pregunta: ¿En qué consiste el trámite Concepto de norma urbanística? **Respuesta deseada:** El trámite Concepto de norma urbanística consiste en: Obtener información sobre la reglamentación urbanística aplicable a un predio específico.

Pregunta: ¿Cuánto tiempo tarda el trámite Concepto de norma urbanística? **Respuesta deseada:** El trámite Concepto de norma urbanística tiene un tiempo de respuesta de 15 Día(s) hábil(es).

Pregunta: ¿El trámite Concepto de norma urbanística es gratuito? **Respuesta deseada:** El trámite Concepto de norma urbanística tiene costo.

Pregunta: ¿Pasos para el trámite Concepto de norma urbanística? **Respuesta deseada:** "Para realizar el trámite Concepto de norma urbanística, sigue estos pasos:",1. Reunir documentos y cumplir condiciones: Documento de identificación del solicitante con Fotocopia(s), Matrícula inmobiliaria del predio con Fotocopia(s), Formulario de solicitud debidamente diligenciado con Original(es),2. Radicar documentos en la ventanilla única,3. Realizar el pago correspondiente,4. Recibir el concepto de norma urbanística

Pregunta: ¿Quién se encarga del trámite Concepto de norma urbanística? **Respuesta deseada:** El organismo responsable del trámite Concepto de norma urbanística es el Departamento Administrativo de Planeación Municipal.

Pregunta: Explícame el trámite Licencia de construcción en la modalidad de cerramiento **Respuesta deseada:** El trámite Licencia de construcción en la modalidad de cerramiento consiste en: Obtener la autorización para encerrar de manera permanente un predio de propiedad privada.

Pregunta: ¿Duración del trámite Licencia de construcción en la modalidad de cerramiento? **Respuesta deseada:** El trámite Licencia de construcción en la modalidad de cerramiento tiene un tiempo de respuesta de 45 Día(s) hábil(es), prorrogable por la mitad del término inicial.

Pregunta: ¿El trámite Licencia de construcción en la modalidad de cerramiento tiene algún costo? **Respuesta deseada:** El trámite Licencia de construcción en la modalidad de cerramiento tiene costo, asociado al impuesto de delineación urbana y posibles expensas.

Pregunta: ¿Cómo puedo realizar el trámite Licencia de construcción en la modalidad de cerramiento? **Respuesta deseada:** "Para realizar el trámite Licencia de construcción en la modalidad de cerramiento, debes:",1. Cumplir con los requisitos documentales y técnicos establecidos por la curaduría urbana,2. Radicar la solicitud ante una de las curadurías urbanas de Cali,3. Realizar los pagos correspondientes a expensas y el impuesto de delineación,4. Una vez aprobada, notificarse y retirar la licencia.

Pregunta: ¿Qué organismo es responsable de la Licencia de construcción en la modalidad de cerramiento? **Respuesta deseada:** El organismo responsable del trámite Licencia de construcción en la modalidad de cerramiento es la Curaduría Urbana (asignada por turno o elección del solicitante).

Pregunta: ¿Qué es el trámite Estratificación socioeconómica de un inmueble residencial? **Respuesta deseada:** El trámite Estratificación socioeconómica de un inmueble residencial consiste en: Solicitar la asignación o modificación de la estratificación socioeconómica de un inmueble residencial.

Pregunta: ¿Cuál es el tiempo de respuesta para el trámite Estratificación socioeconómica de un inmueble residencial? **Respuesta deseada:** El trámite Estratificación socioeconómica de un inmueble residencial tiene un tiempo de respuesta de 2 Mes(es).

Pregunta: ¿El trámite Estratificación socioeconómica de un inmueble residencial tiene costo? **Respuesta deseada:** El trámite Estratificación socioeconómica de un inmueble residencial no tiene costo.

Pregunta: ¿Cuáles son los pasos para solicitar la Estratificación socioeconómica de un inmueble residencial?

Respuesta deseada: "Para el trámite Estratificación socioeconómica de un inmueble residencial:",1. Presentar solicitud escrita indicando dirección del predio y motivo,2. Anexar copia de un recibo de servicio público del inmueble,3. Radicar la solicitud en el Departamento Administrativo de Planeación Municipal.

Anexo B: Preguntas utilizadas en evaluación humana.

Las siguientes 10 preguntas, extraídas del archivo Evaluación cualitativa Modelos (2).xlsx - Consolidado.csv, fueron presentadas a los tres evaluadores expertos para la calificación de las respuestas generadas por los modelos:

1. ¿Cómo puedo modificar el área y los linderos de mi propiedad si estoy de acuerdo con mi vecino?
2. ¿Qué debo hacer para obtener un permiso para ocupar temporalmente el antejardín de mi negocio?
3. Necesito la nomenclatura oficial de mi casa, ¿qué trámite debo seguir?
4. ¿Cuánto tiempo se demora el trámite de asignación de nomenclatura?
5. ¿Tiene algún costo el trámite de asignación de nomenclatura?
6. ¿Qué organismo es el responsable del trámite de asignación de nomenclatura?
7. ¿Dónde puedo realizar el trámite de asignación de nomenclatura?
8. ¿Cuáles son los pasos para el trámite de Permiso para ocupación temporal de antejardines y/o espacio público contiguo al establecimiento de comercio?
9. ¿Quién se encarga del trámite de Permiso para ocupación temporal de antejardines y/o espacio público contiguo al establecimiento de comercio?
10. ¿Qué costo tiene el trámite de Permiso para ocupación temporal de antejardines y/o espacio público contiguo al establecimiento de comercio?

Anexo C. Otros Hiperparámetros usados en el entrenamiento de los modelos Phi y Llama

Este anexo muestra algunas de las diferentes combinaciones de hiperparámetros utilizados para el entrenamiento de los modelos en la fase de exploración, antes de identificar el mejor conjunto expuesto en el capítulo 7, en comparación con la mejor configuración en estos ejemplos se observa principalmente que los ajustes se encuentran en el número de épocas menor para Llama y el uso de pasos en su lugar (max_steps) para Phi, así como y una tasa mayor de aprendizaje (learning_rate), estos dos hiperparámetros que se identificaron como críticos, fueron los que generaron el mayor impacto en los resultados obtenidos por los modelos en las evaluaciones cuantitativas y cualitativas posterior a la ampliación de la data.

Ejemplo de Hiperparámetros utilizados para el Modelo Llama 3.2 3B Instruct

Hiperparámetro de SFTTrainer / TrainingArguments	Valor Configurado	Descripción
dataset_num_proc	2	Número de procesos utilizados para el preprocesamiento de datos (tokenización, mapeo de funciones). Un valor > 1 acelera esta etapa.
gradient_accumulation_steps	8	Número de pasos durante los cuales se acumulan gradientes antes de una actualización de pesos. Permite simular un tamaño de lote efectivo mayor ($\text{per_device_train_batch_size} * \text{gradient_accumulation_steps}$). Batch Efectivo: 32.
warmup_steps	100	Número de pasos iniciales donde la tasa de aprendizaje aumenta linealmente desde 0 hasta learning_rate. Ayuda a estabilizar el entrenamiento temprano.
num_train_epochs	15	Número total de veces que el conjunto de datos de entrenamiento completo se presenta al modelo.
learning_rate	2e-4 (0.0002)	Tasa de aprendizaje inicial para el optimizador. Un valor pequeño y común para fine-tuning.
logging_steps	10	Frecuencia (en pasos) con la que se registran las métricas de entrenamiento (ej. pérdida).
optim	"adamw_8bit"	Optimizador AdamW con estados cuantizados a 8 bits, reduciendo el uso de memoria.
weight_decay	1	Coefficiente de regularización L2 para penalizar pesos grandes y prevenir el sobreajuste.

Ejemplo de Hiperparámetros utilizados para el Modelo Phi 3.5 Mini instruct

Hiperparámetro de SFTTrainer / TrainingArguments	Valor Configurado	Descripción
dataset_num_proc	2	Número de procesos utilizados para el preprocesamiento de datos (tokenización, mapeo de funciones). Un valor > 1 acelera esta etapa.
gradient_accumulation_steps	4	Número de pasos durante los cuales se acumulan gradientes antes de una actualización de pesos. Permite simular un tamaño de lote efectivo mayor ($\text{per_device_train_batch_size} * \text{gradient_accumulation_steps}$). Batch Efectivo: 8.
warmup_steps	5	Número de pasos iniciales donde la tasa de aprendizaje aumenta linealmente desde 0 hasta learning_rate. Ayuda a estabilizar el entrenamiento temprano.
max_steps	60	Número máximo de pasos de entrenamiento a realizar. El entrenamiento se detendrá aquí, independientemente de num_train_epochs (que no está especificado).
learning_rate	2e-4 (0.0002)	Tasa de aprendizaje inicial para el optimizador.
logging_steps	1	Frecuencia (en pasos) con la que se registran las métricas de entrenamiento (ej. pérdida).
optim	"adamw_8bit"	Optimizador AdamW con estados cuantizados a 8 bits, reduciendo el uso de memoria.
weight_decay	0.01	Coefficiente de regularización L2 para penalizar pesos grandes y prevenir el sobreajuste.

11. REFERENCIAS

- [1] S. P. Ñañez, "Informe de seguimiento servicio al ciudadano enero-marzo 2024, Alcaldía Distrital de Santiago de Cali 2024", *Inf. Trimest.*, pp. 6–31, 2024..
- [2] "Quienes somos". Sitio web de la Función Pública. Accedido el 10 de junio de 2024. [En línea]. Disponible: <https://www1.funcionpublica.gov.co/quienes-somos/funciones-generales>
- [3] "Glosario". Sitio web de la Función Pública. Accedido el 10 de junio de 2024. [En línea]. Disponible: <https://www1.funcionpublica.gov.co/web/suit/glosario1>
- [4] "¿Cómo diferenciar un trámite de otros procedimientos administrativos (OPA)?" Sitio web de la Función Pública. Accedido el 10 de junio de 2024. [En línea]. Disponible: <https://www1.funcionpublica.gov.co/web/suit/40>
- [5] D. Jurafsky and J. H. Martin, *Speech and Language Processing*, 3rd ed. draft, 2023. [Online]. Disponible: <https://web.stanford.edu/~jurafsky/slp3/>
- [6] S. Minaee, T. Mikolov, N. Nikzad, M. Chenaghlu, E. Ansari, R. Socher, X. Amatriain, and J. Gao, "Large Language Models: A Survey," *arXiv preprint arXiv:2402.06196*, 2024.
- [7] S. Raschka, *Build a Large Language Model (From Scratch)*. Manning, 2024.
- [8] A. Vaswani et al., "Attention is All You Need," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, 2017, pp. 5998-6008.
- [9] A. Al-Rfou, D. A. Al-lahham, W. K. Al-Shqairat, and M. A. Al-kabi, "Parameter-Efficient Fine-Tuning for Large Models: A Comprehensive Survey," *arXiv preprint arXiv:2403.14608*, 2024.
- [10] J. Kaplan et al., "Scaling Laws for Neural Language Models," *arXiv preprint arXiv:2001.08361*, 2020.
- [11] J. Hoffmann et al., "Training Compute-Optimal Large Language Models," *arXiv preprint arXiv:2203.15556*, 2022.
- [12] Z. Tan, A. Beigi, S. Wang, R. Guo, A. Bhattacharjee, B. Jiang, M. Karami, J. Li, L. Cheng, and H. Liu, "Large Language Models for Data Annotation: A Survey," *arXiv preprint arXiv:2402.13446*, 2024.
- [13] "What is Catastrophic Forgetting?," IBM. [En línea]. Disponible: <https://www.ibm.com/think/topics/catastrophic-forgetting>. [Accessed: May 17, 2025].
- [14] Y. Gao, Y. Xiong, X. Gao, K. Jia, J. Pan, Y. Bi, Y. Dai, J. Sun, and M. Guo, "Retrieval-Augmented Generation for Large Language Models: A Survey," *arXiv preprint arXiv:2312.10997*, 2023.
- [15] A. M. El-Sayed, M. I. Khalil, and H. M. Abbas, "A Comprehensive Review of Retrieval-Augmented Generation (RAG): Key Challenges and Future Directions," *arXiv preprint arXiv:2410.12837*, 2024.

- [16] A. Bandyopadhyay, "RAG vs. Fine tuning: Which AI strategy should you choose?," *IBM Developer*, Dic. 03, 2024. [En línea]. Disponible: <https://developer.ibm.com/articles/awb-rag-vs-fine-tuning/>. [Accessed: May 17, 2025].
- [17] L. P. Kaelbling, M. L. Littman, and A. W. Moore, "Reinforcement Learning: A Survey," *Journal of Artificial Intelligence Research*, vol. 4, pp. 237-285, 1996.
- [18] "Fine-Tuning LLMs for Industry Success: Effective Data Labeling Techniques," Hyqoo. [En línea]. Disponible: <https://hyqoo.com/technical/fine-tuning-llms-for-industry-success-effective-data-labeling-techniques/>. [Accessed: May 17, 2025].
- [19] "Data labeling: a practical guide (2024)," Snorkel AI, Sep. 29, 2023. [En línea]. Disponible: <https://snorkel.ai/data-labeling/>. [Accessed: May 17, 2025].
- [20] "Evaluation for finetuning," ZenML Docs. [En línea]. Disponible: <https://docs.zenml.io/user-guides/llmops-guide/finetuning-llms/evaluation-for-finetuning>. [Accessed: May 17, 2025].
- [21] C. Zhou, P. Liu, P. Si, Y. Mao, G. Neubig, A. W. Yu, and V. Stoyanov, "LIMA: Less Is More for Alignment," *arXiv preprint arXiv:2305.11206*, 2023.
- [22] S. Raschka, "Optimizing LLMs From a Dataset Perspective," *Sebastian Raschka's Blog*, Sep. 15, 2023. [En línea]. Disponible: <https://sebastianraschka.com/blog/2023/optimizing-LLMs-dataset-perspective.html>. [Accessed: May 17, 2025].
- [23] "Decoding the Challenges of LLM Evaluation and Tools to Solve Them," InsightFinder, Ene. 15, 2025. [En línea]. Disponible: <https://insightfinder.com/blog/why-evaluating-large-language-models-critical/>. [Accessed: May 17, 2025].
- [24] "The Rise of Large Language Models in Automatic Evaluation: Why We Still Need Humans in the Loop," Thomson Reuters, Ene. 21, 2025. [En línea]. Disponible: <https://www.thomsonreuters.com/en-us/posts/innovation/the-rise-of-large-language-models-in-automatic-evaluation-why-we-still-need-humans-in-the-loop/>. [Accessed: May 17, 2025].
- [25] "A Comprehensive Guide to LLM Alignment and Safety," Turing. [En línea]. Disponible: <https://www.turing.com/resources/llm-alignment-and-safety-guide>. [Accessed: May 17, 2025].
- [26] "LLM Evaluation Metrics Every Developer Should Know," Comet.ml, Ene. 3, 2025. [En línea]. Disponible: <https://www.comet.com/site/blog/llm-evaluation-metrics-every-developer-should-know/>. [Accessed: May 17, 2025].
- [27] S. Raschka, *Machine Learning Q and AI*. San Francisco, CA: No Starch Press, 2024.