Maestría en Ciencia de Datos
Facultad de Ingeniería y Ciencias

FICHA RESUMEN
TRABAJO DE GRADO DE MAESTRÍA

TITULO: "Clasificación de reportes de daños realizados por los canales digitales de Celsia mediante Agentes Virtuales"

1. ÉNFASIS: N/A
2. TIPO DE PROYECTO: Aplicado
3. ÁREA DE TRABAJO: Ciencia de Datos Aplicada a Operaciones Eléctricas
4. ESTUDIANTE (S): María Fernanda Hernández Porras y Jesús David Sandoval Posso
5. CORREO ELECTRÓNICO: mariahernandez96@javerianacali.edu.co y jdposso@javerianacali.edu.co
6. DIRECCIÓN Y TELÉFONO: Cll 60ª#107-104 | 3167586829; Cll 15ª # 69-85 | 3166227535
7. DIRECTOR: Juan Carlos González Vélez
8. VINCULACIÓN DEL DIRECTOR (en la universidad): Externo
9. CORREO ELECTRÓNICO DEL DIRECTOR: jcgonzalezv@celsia.com
10. CO-DIRECTOR(ES): No Aplica
11. GRUPO O EMPRESA QUE LO AVALA: Celsia Colombia SA ESP
12. OTROS GRUPOS O EMPRESAS: No Aplica
13. PALABRAS CLAVE (al menos 5): Agente virtual, priorización de incidentes, NLP, energía eléctrica, eficiencia operativa, Design Thinking, LangChain, LangGraph, CrewAI
14. ODS QUE APLICA EL PROYECTO (Agenda 2030): ODS 7 (Energía Asequible y No Contaminante), ODS 9 (Industria, Innovación e Infraestructura) y ODS 11 (Ciudades y Comunidades Sostenibles)
15. FECHA DE INICIO (Desarrollo del proyecto): 1/06/2025
16. RESUMEN: En Celsia Colombia S.A. E.S.P., la clasificación manual e inadecuada de reportes de incidentes recibidos por canales digitales, como el sistema LuzIA, genera un alto número de visitas técnicas fallidas, lo que incrementa los tiempos de respuesta, reduce la eficiencia operativa y afecta la experiencia del cliente. Sobre una muestra de 88.959 registros de incidentes cancelados históricos del período comprendido entre 2023–2025, el análisis exploratorio identificó 5.111 visitas fallidas, equivalentes al 5,75% de los casos cancelados, concentradas en reportes mal clasificados o canalizados incorrectamente. La deficiencia en los filtros y criterios de priorización homogeniza la atención de incidentes, causando demoras en casos críticos, aumentando los indicadores SAIDI y SAIFI, y exponiendo a la empresa a sanciones regulatorias. Este proyecto, desarrollado en el marco de la Maestría en Ciencia de Datos de la Pontificia Universidad Javeriana Cali, propone un sistema de agentes virtuales basado en frameworks como LangChain, LangGraph y CrewAI para detectar, validar y clasificar automáticamente reportes, diferenciando entre daños en infraestructura y pérdida de servicio. La solución busca optimizar la asignación de recursos, reducir visitas fallidas y mejorar la experiencia del cliente en Valle del Cauca y Tolima, demostrando su viabilidad mediante un prototipo funcional que promueve la agilidad, confiabilidad e innovación en la gestión de incidentes eléctricos.

CLASIFICACIÓN DE REPORTES DE DAÑOS REALIZADOS POR LOS CANALES DIGITALES DE CELSIA MEDIANTE AGENTES VIRTUALES

Jesús David Sandoval Posso; María Fernanda Hernández Porras

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

Director
PhD(C). Juan Carlos González Vélez

FACULTAD DE INGENIERÍA Y CIENCIAS
MAESTRÍA EN CIENCIA DE DATOS
SANTIAGO DE CALI, MAYO 25 DE 2026

TABLA DE CONTENIDO

LISTA DE ILUSTRACIONES

LISTA DE TABLAS

LISTA DE ANEXOS

1. INTRODUCCIÓN

La energía eléctrica constituye un pilar fundamental en el desarrollo económico y social de cualquier nación. En Colombia, el sector eléctrico provee cobertura al casi 96% de la población, y su demanda crece de forma sostenida impulsada por la urbanización, la mayor electrificación industrial y la incorporación de nuevas tecnologías. Esta expansión convierte al sector eléctrico en un escenario de alta complejidad operativa, ya que cualquier falla en la red se traduce de forma inmediata en una interrupción del servicio. Hospitales, fábricas, comercios, hogares y sistemas de telecomunicaciones dependen sin excepción del suministro continuo de energía eléctrica, y una interrupción, por más breve que sea, puede llegar a ocasionar pérdidas económicas, riesgos de seguridad y deterioro en la calidad de vida de las personas. Por ello, cuando se presenta una falla, la prioridad operativa es identificar el daño, despachar los recursos técnicos adecuados y restablecer el servicio en el menor tiempo posible.

Esta exigencia no es solo operativa, sino también regulatoria. La Comisión de Regulación de Energía y Gas (CREG), mediante la Resolución 015 de 2018, estableció el marco de calidad del servicio que rige a los Operadores de Red del país, en torno a dos indicadores de clase mundial que son el SAIDI (System Average Interruption Duration Index), que mide cuántas horas al año estuvo sin servicio un usuario, y el SAIFI (System Average Interruption Frequency Index), que mide cuántas veces al año se vio interrumpido su suministro de energía. Las metas de largo plazo fijadas por la CREG son parámetros que la Superintendencia de Servicios Públicos Domiciliarios (SSPD) vigila constantemente y que sirven de base para el esquema de incentivos, compensaciones y eventuales sanciones regulatorias. Cuando una brigada técnica sale innecesariamente a campo, porque un reporte fue mal clasificado y no correspondía a un daño real en la red, se produce una consecuencia en cadena, los recursos se consumen en una visita fallida, los usuarios con interrupciones genuinas esperan más tiempo, las horas sin servicio presionan el SAIDI y el número de eventos sin resolver a tiempo deteriora el SAIFI. A esto se suman costos operativos directos, como lo son combustible, horas-hombre, desgaste de vehículos e indirectos que comprometen el presupuesto disponible para mantenimiento preventivo y mejora de infraestructura.

Es precisamente en este cruce entre complejidad operativa y presión regulatoria donde se inscribe el presente proyecto. Celsia Colombia S.A. E.S.P. es una empresa distribuidora y comercializadora del servicio de energía eléctrica en los departamentos del Valle del Cauca y de Tolima. Su canal digital LuzIA gestiona actualmente el 93% de los reportes sin intervención humana y al corte de marzo de 2026 acumulaba más de 1.004.292 conversaciones en WhatsApp, con un promedio mensual de 83.707 interacciones. A pesar de este volumen de atención automatizada, la empresa enfrenta un desafío crítico: el 9,6% de esas conversaciones permanece sin clasificar correctamente, generando un número significativo de visitas técnicas fallidas. Los flujos de atención, aunque capturaban datos básicos del cliente y del problema, carecían del diagnóstico necesario para distinguir si un reporte correspondía a un daño en infraestructura de red, a un problema interno de la instalación del cliente, o a una tipología diferente como alumbrado público, facturación o reconexión. Esta ambigüedad propiciaba el ciclo de visitas

fallidas, reprocesos y deterioro de los indicadores de calidad en los municipios con mayor concentración de reportes: Ibagué, Espinal, Palmira y Melgar.

Desde la perspectiva de la ciencia de datos, el reto no radica en la falta de información: todos los reportes quedan registrados como texto libre en la base de datos del sistema ADMS, conformando un historial valioso que incluye el resultado final de cada visita. El verdadero desafío está en que esa información no se aprovecha plenamente para resolver la brecha residual que el sistema actual no ha logrado cerrar. El 9,6% de conversaciones sin clasificar y las tipologías ajenas que ingresan al flujo de energía representan una oportunidad concreta para aplicar técnicas de procesamiento de lenguaje natural, embeddings semánticos y clasificación supervisada que permitan identificar patrones, refinar la discriminación entre tipologías y reducir las visitas técnicas fallidas antes del despacho. Entre los desafíos técnicos que esta aproximación implica se encuentran el preprocesamiento del texto informal de los clientes, el desbalance extremo entre clases del dataset histórico, la definición de umbrales de confianza y el diseño de una arquitectura que se integre al sistema existente sin reemplazarlo.

Para abordar esta problemática, se desarrolló un prototipo de agentes virtuales capaz de guiar la conversación con el usuario, recoger datos diagnósticos clave como si el problema afecta solo a su vivienda o también a sus vecinos, si percibe parpadeo en las luces o escucha zumbido en el transformador cercano y clasificar automáticamente cada reporte entre las tipologías operativas de Celsia, como paso previo a la creación del ticket en el sistema de gestión de incidentes (ADMS). El sistema se construyó sobre los frameworks LangChain, LangGraph y CrewAI, combinados con técnicas de Procesamiento de Lenguaje Natural (NLP) para interpretar el texto en español enviado por los usuarios. La metodología se estructuró en tres etapas: un ejercicio de Design Thinking con operadores y brigadas para co-diseñar el flujo conversacional; la construcción del sistema de agentes especializados apoyado en un grafo de conocimiento; y la validación del prototipo en un entorno controlado, midiendo precisión de clasificación, tiempos de respuesta y percepción de usabilidad por parte de usuarios de prueba.

El documento se organiza de la siguiente manera: el capítulo 2 presenta la contextualización del proyecto, incluyendo la definición del problema, los objetivos y el marco de referencia con los fundamentos teóricos sobre NLP, bases de datos vectoriales y diseño centrado en el usuario. El capítulo 3 desarrolla el diseño del flujo conversacional centrado en el cliente. El capítulo 4 describe la construcción del sistema inteligente de clasificación de reportes. El capítulo 5 presenta la validación del agente virtual y la evaluación de su desempeño operativo. Finalmente, el capítulo 6 recoge las conclusiones, los trabajos futuros y las referencias bibliográficas que sustentan el estudio. Este proyecto, enmarcado en la Maestría en Ciencia de Datos de la Pontificia Universidad Javeriana Cali, aplica ciencia de datos al sector eléctrico con el objetivo de mejorar la operación de Celsia y la experiencia de sus clientes, contribuyendo a los valores que orientan a la empresa, simplicidad, agilidad, confiabilidad e innovación.

2. CONTEXTUALIZACION DEL PROYECTO

2.1 DEFINICIÓN DEL PROBLEMA

2.1.1 PLANTEAMIENTO DEL PROBLEMA

En Celsia, el proceso actual de recepción y clasificación de reportes de incidentes presenta deficiencias que impactan la correcta y eficaz operativa y la satisfacción del cliente. A continuación, se describen las principales causas y efectos del problema central detectados en el flujo de atención al cliente.

Si bien a partir de agosto de 2025 la migración hacia un sistema basado en modelos de lenguaje grande (LLM) elevó la efectividad de clasificación hasta aproximadamente el 95%, persiste aún un 9,6% de conversaciones que al corte de marzo de 2026 no logran clasificarse correctamente, y una fracción de reportes ajenos al servicio eléctrico que continúan generando despachos innecesarios de brigadas. Son estas dos brechas residuales las que motivan el presente proyecto.

Los flujos de atención en los canales digitales de Celsia fueron diseñados para recopilar información básica sobre ubicación y contacto del usuario, pero no incluyen preguntas específicas que permitan distinguir con certeza si un reporte corresponde a una falla real del suministro eléctrico o a otro tipo de solicitud relacionada con su infraestructura. Al carecer de criterios claros desde el primer momento, se registran reportes con datos incompletos y/o confusos, lo que en ocasiones obliga a realizar inspecciones en campo para determinar la naturaleza del incidente. Esta carencia ralentiza la atención y es una de las brechas que el presente proyecto busca cerrar.

Aunque el sistema actual clasifica correctamente el 90,4% de las conversaciones, el porcentaje restante sin clasificar, sumado a tipologías ajenas al servicio eléctrico, como aseo, alumbrado público, reclamaciones, reconexiones y clientes en zonas de peaje, que continúan llegando al flujo de daños de energía, provoca que los incidentes críticos compitan en la lista de pendientes con reportes que no corresponden a daños reales en la red. Como resultado, las brigadas técnicas deben esperar su turno, extendiendo los tiempos de interrupción para un gran número de clientes y reduciendo la percepción de confiabilidad del servicio.

Dado que persiste esta brecha en la clasificación, una fracción de las órdenes de despacho se sigue elaborando con datos incompletos o sobre reportes mal canalizados. Con frecuencia, las brigadas llegan al lugar indicado y descubren que no hay un corte de red ni un daño en infraestructura; puede tratarse de un problema interno de la instalación del usuario, una solicitud de reconexión o un servicio ajeno al portafolio de Celsia. El análisis exploratorio realizado sobre una muestra de 88.959 registros históricos de ADMS del período 2023–2025, con datos provistos directamente por Celsia, identificó 5.111 visitas fallidas, equivalentes al 5,75% del total de casos cancelados. Las tipologías más frecuentes fueron podas (23,5%), dato insuficiente (18,8%) y falla interna del cliente (17,6%). Cada visita fallida representa un costo estimado de $90.000 pesos colombianos según información del equipo operativo de Celsia, cifra que varía según el contratista asignado. Proyectando este valor sobre las 5.111 visitas identificadas, el impacto

económico acumulado supera los $459 millones de pesos sin incluir costos indirectos, comprometiendo el presupuesto disponible para mantenimiento preventivo y mejora de infraestructura.

El equipo de brigadas en el Valle del Cauca y Tolima dispone de un número limitado de técnicos especializados. La asignación de personal a incidentes derivados de clasificaciones erróneas reduce su disponibilidad para atender cortes de alta criticidad, generando retrasos que incrementan las reclamaciones e interrupciones prolongadas. Este efecto se refleja directamente en los indicadores SAIDI y SAIFI, que miden respectivamente la duración y frecuencia promedio de las interrupciones en el suministro eléctrico. Cuando la atención se retrasa por una priorización inadecuada, estos índices se elevan, lo que no solo compromete la satisfacción del cliente sino que puede derivar en sanciones regulatorias y daño reputacional ante las entidades de control.
A continuación, en la ilustración 1, se observa de forma interactiva el flujo que actualmente tiene el procedimiento:

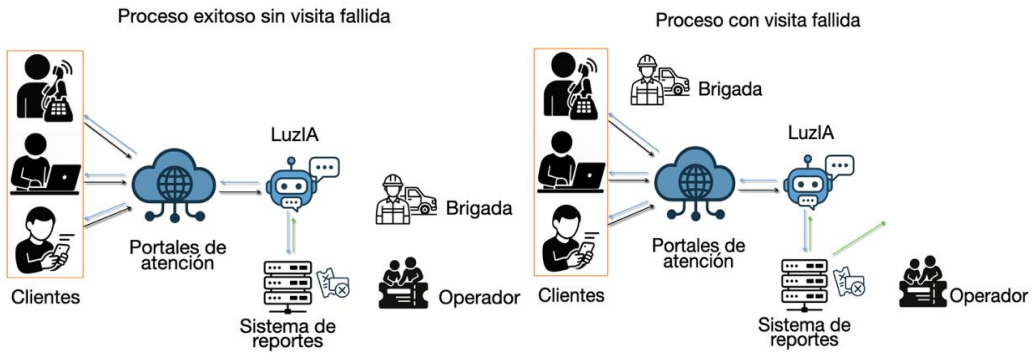


Ilustración 1. Flujo de portales web. Caso de proceso exitoso sin visita fallida y caso de proceso con visita fallida.

A la izquierda, se observa que, gracias a una rápida intervención del operador, aunque el portal web generó un tiquete de daño basado en el reporte del cliente, el operador, al analizarlo, determinó que no requiere la atención de las brigadas de daños para resolver el problema. Por otro lado, en la imagen de la derecha, se nota que, debido a la alta carga operativa, en varias ocasiones el operador no logra analizar el tiquete de daños creado por los portales de manera oportuna tras un reporte del cliente, lo que resulta en el despacho de la brigada de daños al sitio. Posteriormente, la misma operativa identifica que el reporte no correspondía a un daño real.

Para abordar estas dificultades, se propone implementar un sistema de cuatro agentes virtuales especializados sobre *LangGraph* y *CrewAI*, construido sobre la infraestructura LLM existente, que clasifique semánticamente el porcentaje de conversaciones no resueltas, filtre los reportes que no corresponden al flujo de energía antes de generar eventos en el sistema ADMS. Con ello se busca reducir las visitas fallidas en un rango estimado del 40% al 70%, optimizar el uso de los recursos técnicos en campo y contribuir a la mejora de los indicadores de calidad del servicio, fortaleciendo la operatividad y la reputación de Celsia.

(Nota: los registros utilizados como evidencia cuantitativa son de fuente interna de Celsia y están disponibles para el evaluador bajo solicitud, sujeto a las condiciones de confidencialidad acordadas con la empresa.)

2.1.2 FORMULACIÓN DEL PROBLEMA

¿En qué media un sistema de agentes virtuales basado en *LangGraph* y *CrewAI* puede automatizar la clasificación multiclase de los reportes de daños recibidos por LuzIA, diferenciando entre fallas reales en la red eléctrica y solicitudes de otro tipo, con un nivel de precisión operacionalmente aceptable que permita reducir entre un 40% y un 70% las visitas técnicas fallidas y optimizar la asignación de recursos en campo en Celsia Colombia S.A. E.S.P.?

2.1.3 SISTEMATIZACIÓN DEL PROBLEMA

Para sistematizar el problema, se descompone en las siguientes preguntas específicas:

1. ¿Cómo debe estructurarse el flujo conversacional del agente virtual para guiar al cliente en la recolección de información relevante que permita diferenciar adecuadamente entre eventos de pérdida de servicio y fallas en infraestructura?

2. ¿Qué modelos de clasificación y herramientas basadas en inteligencia artificial son más adecuados para categorizar automáticamente los reportes de incidentes según su tipo, garantizando precisión y eficiencia operativa?

3. ¿Qué métricas y técnicas de validación permiten evaluar el desempeño del agente virtual en la clasificación automática de reportes, y cómo se refleja su impacto en la reducción de visitas fallidas y en la mejora de la experiencia del cliente?

2.2 OBJETIVOS DEL PROYECTO

2.2.1 OBJETIVO GENERAL

Diseñar una metodología basada en agentes virtuales que clasifique el tipo de daño reportado por los clientes de Celsia a través de los canales de atención digital, diferenciando entre eventos relacionados con infraestructura y pérdida de servicio, con el fin de optimizar la asignación de recursos técnicos, reducir las visitas fallidas en campo y mejorar la eficiencia operativa y la experiencia del cliente.

2.2.2 OBJETIVOS ESPECÍFICOS

Objetivo específico 1. Diseñar un flujo de interacción centrado en el cliente, utilizando técnicas

de *Design Thinking*, que optimice la recolección de información y facilite la categorización inicial de los reportes de incidentes según el tipo de daño.

Objetivo específico 2. Construir un sistema inteligente de clasificación de reportes de incidentes, implementando flujos de agentes virtuales basados en las herramientas *LangGraph* y *CrewAI*, para garantizar una priorización eficiente y precisa.

Objetivo específico 3. Validar el funcionamiento del agente virtual en el canal de atención al cliente mediante técnicas de validación cruzada, asegurando su correcto desempeño y fiabilidad en la clasificación de reportes.

2.3 MARCO DE REFERENCIA

2.3.1 MARCO TEÓRICO

Experiencia al Cliente y su Optimización con IA

La experiencia del cliente (CX) es actualmente uno de los factores más determinantes en la competitividad empresarial, especialmente en sectores de servicios como el eléctrico. La inteligencia artificial (IA) ha revolucionado la CX al permitir el análisis de grandes volúmenes de datos provenientes de múltiples canales de interacción, tales como chats, correos electrónicos, redes sociales y llamadas telefónicas [1]. Este análisis posibilita la identificación de patrones de comportamiento, preferencias y necesidades individuales, lo que facilita la personalización de la atención y la anticipación de requerimientos futuros [2][3]. Empresas líderes como Rappi en Colombia han logrado incrementar el *engagement* y la retención de clientes gracias a la implementación de IA en la personalización de recomendaciones y la optimización de la experiencia de usuario [4].

La personalización, como elemento clave de la CX, se logra principalmente mediante el uso de datos históricos y en tiempo real de cada cliente. Según Microsoft el 72% de los consumidores esperan que los agentes de servicio al cliente conozcan su historial de compras e interacciones previas, lo que subraya la importancia de contar con sistemas integrados de gestión de información [2]. La IA no solo permite responder rápidamente a las consultas, sino que también contribuye a la detección temprana de problemas y la mejora continua de los procesos operativos. Además, el análisis de emociones basado en IA ayuda a identificar el estado de ánimo de los clientes durante la interacción, permitiendo ajustar la estrategia de atención y aumentar los niveles de satisfacción y fidelización [2][3].

En Colombia, la percepción de los usuarios frente a la IA en la atención al cliente es positiva cuando se logra un equilibrio adecuado entre la automatización y el trato humano. Estudios recientes muestran que los consumidores valoran especialmente la rapidez y precisión de las respuestas automatizadas, pero también requieren la posibilidad de ser atendidos por un agente humano en casos complejos o sensibles [3][19].

Machine Learning, Deep Learning y NLP

El *machine learning* (ML) y el *deep learning* (DL) son pilares fundamentales en la automatización y mejora de procesos complejos dentro de la inteligencia artificial. El ML permite que los sistemas aprendan de datos históricos para tomar decisiones, clasificar información y predecir comportamientos futuros, mientras que el DL utiliza redes neuronales profundas para identificar patrones complejos en grandes volúmenes de datos no estructurados, como imágenes, texto y audio [5][6]. Estas técnicas han permitido avances significativos en la atención al cliente, donde los *chatbots* inteligentes pueden manejar miles de interacciones simultáneas, aprendiendo de cada una para mejorar su rendimiento y precisión [6].

El NLP es una de las aplicaciones más relevantes de ML y DL en el contexto de la experiencia del cliente. El NLP facilita la comprensión, interpretación y generación de lenguaje humano, permitiendo a los sistemas analizar reportes de clientes, clasificar automáticamente solicitudes y detectar emociones en el texto [7][6].

Modelos avanzados como BERT y sus variantes específicas para el español, como BETO y BERTin, han demostrado una alta efectividad en tareas de clasificación, extracción de información y análisis de sentimientos, lo que los hace ideales para aplicaciones en atención al cliente y gestión de incidentes [7]. Además, la capacidad de estos modelos para adaptarse a diferentes dialectos y contextos lingüísticos los convierte en herramientas versátiles para empresas multinacionales o con presencia en regiones con variantes lingüísticas.

La integración de ML, DL y NLP en los sistemas de atención al cliente no solo mejora la eficiencia operativa, sino que también permite ofrecer respuestas más relevantes y personalizadas a los usuarios. Por ejemplo, el análisis de sentimientos en tiempo real ayuda a las empresas a identificar clientes insatisfechos y actuar proactivamente para resolver sus problemas, mejorando la reputación y la fidelización [6][7]. Asimismo, la automatización de tareas repetitivas libera a los agentes humanos para que se enfoquen en casos complejos o sensibles, optimizando así el uso de recursos y elevando la calidad del servicio [6].

Diseño y Desarrollo de Software: Fases y Metodologías

El diseño y desarrollo de software para agentes virtuales es un proceso complejo que integra disciplinas como la inteligencia artificial, la ingeniería de software y la interacción humano-computadora. Un agente virtual se define como una entidad de software capaz de percibir su entorno, procesar información y ejecutar acciones para alcanzar objetivos específicos, ya sea de manera autónoma o en respuesta a estímulos externos. Estos agentes son fundamentales en entornos virtuales inteligentes y pueden comunicarse con otros agentes para intercambiar información y coordinar acciones, lo que les permite ejecutar tareas de manera eficiente y adaptativa [9].

El ciclo de vida del desarrollo de software para agentes virtuales suele estructurarse en varias fases clave: recolección y análisis de requisitos, diseño del sistema, implementación, pruebas y despliegue. Durante la fase de análisis, se identifican las necesidades del usuario y los objetivos funcionales del agente. El diseño implica la definición de la arquitectura, los modelos de comportamiento y las interacciones con el entorno y otros agentes. La implementación transforma estos diseños en código, utilizando lenguajes y *frameworks* adecuados para garantizar la escalabilidad y mantenibilidad del sistema. Posteriormente, se llevan a cabo pruebas exhaustivas —unitarias, de integración y de aceptación— para asegurar la calidad y el correcto funcionamiento del agente antes de su puesta en producción [9][10].

En cuanto a metodologías, el desarrollo de agentes virtuales puede beneficiarse tanto de enfoques tradicionales como de metodologías específicas para sistemas multi-agente. Modelos como Waterfall (Cascada) ofrecen un proceso secuencial y controlado, útil cuando los requisitos están bien definidos, mientras que metodologías ágiles e incrementales permiten mayor flexibilidad y adaptación a cambios durante el desarrollo [8][22]. Para sistemas multi-agente, existen metodologías especializadas como MaSE, ZEUS e INGENIAS, que estructuran el proceso en etapas como análisis del dominio, diseño de agentes, realización e integración, y soporte en tiempo de ejecución. Estas metodologías proporcionan herramientas y marcos conceptuales para modelar roles, comunicaciones y arquitecturas de agentes, facilitando un desarrollo disciplinado y repetible de agentes virtuales sofisticados [23].

***Design Thinking*: Definición y Aplicación**

El *Design Thinking* es una metodología de innovación centrada en el usuario que ha demostrado ser altamente efectiva en la resolución de problemas complejos y la generación de soluciones creativas. Su enfoque principal está en comprender profundamente las necesidades, motivaciones y desafíos de los usuarios, lo que se logra a través de técnicas como entrevistas, observaciones y estudios etnográficos durante la fase de empatía [11][12]. Esta comprensión profunda permite definir el problema de manera precisa y relevante, sentando las bases para la generación de ideas innovadoras en la fase de ideación, donde se fomenta el pensamiento divergente y la colaboración interdisciplinaria [11][12].

La fase de prototipado es fundamental para materializar las ideas y validarlas rápidamente con los usuarios. Los prototipos pueden ir desde bocetos y maquetas hasta versiones digitales funcionales, permitiendo obtener retroalimentación temprana y realizar ajustes antes de invertir recursos significativos en el desarrollo final [11][12]. La fase de prueba consiste en evaluar los prototipos con usuarios reales, identificar posibles mejoras y refinar la solución hasta alcanzar un producto que satisfaga plenamente las necesidades identificadas. Este proceso iterativo garantiza que la solución final sea relevante, usable y deseable para el usuario [11][12].

En Colombia, el *Design Thinking* ha sido adoptado por empresas de diversos sectores, desde servicios financieros hasta tecnología y salud, demostrando su capacidad para impulsar la

innovación y la transformación digital [62][63]. Ejemplos como Bancolombia y su área de Innovación, donde se han desarrollado más de 30 proyectos con impacto tangible, evidencian la efectividad de esta metodología en el fortalecimiento de procesos y la creación de productos centrados en el usuario [62]. Además, el *Design Thinking* ha sido clave en el éxito de startups y microempresas, facilitando la adaptación a contextos cambiantes y la generación de valor diferenciado en el mercado [62][63].

Modelos de Lenguaje de Gran Tamaño (LLM) y Arquitectura *Transformer*

Los modelos de lenguaje de gran tamaño representan un avance revolucionario en el campo del procesamiento de lenguaje natural y la inteligencia artificial. Estos modelos, como GPT y BERT, están entrenados con enormes volúmenes de datos textuales y basan su funcionamiento en la arquitectura *Transformer*, que utiliza mecanismos de autoatención para procesar secuencias de texto de manera eficiente y paralela [14][16]. La arquitectura *Transformer* consta de dos componentes principales: el codificador, que transforma la entrada en una representación vectorial, y el decodificador, que genera la salida a partir de esa representación [7][16]. Esta estructura permite a los LLM comprender el contexto, las relaciones semánticas y la sintaxis del lenguaje, superando las limitaciones de modelos anteriores como las redes recurrentes (RNN) y las redes de memoria a largo plazo (LSTM) [7][16].

El pre-entrenamiento y el *fine-tuning* son fases clave en el desarrollo de los LLM. Durante el pre-entrenamiento, el modelo aprende a predecir palabras faltantes en una frase, adquiriendo un conocimiento general del lenguaje y la estructura gramatical [15][16]. Posteriormente, en la fase de *fine-tuning*, el modelo se adapta a tareas específicas, como clasificación de texto, análisis de sentimientos o generación de respuestas, mediante el entrenamiento con datos etiquetados relevantes para la aplicación deseada [7][16]. Esta capacidad de adaptación ha permitido que los LLM sean utilizados en una amplia variedad de aplicaciones, desde *chatbots* y asistentes virtuales hasta motores de búsqueda y sistemas de recomendación [14][16].

Vectorización y Bases de Datos Vectoriales

La vectorización es el proceso de convertir datos no estructurados, como texto, imágenes o audio, en representaciones numéricas (vectores) que pueden ser procesadas por algoritmos de *machine learning* y *deep learning* [17][18]. Técnicas como *Word2Vec*, *FastText* y *embeddings* de Transformers permiten capturar relaciones semánticas y contextuales entre palabras y frases, lo que es fundamental para tareas de clasificación, búsqueda y recomendación [17][19]. La vectorización facilita el análisis y la comparación de grandes volúmenes de datos, permitiendo a los sistemas identificar patrones y similitudes que no serían evidentes mediante métodos tradicionales [17][18].

Las bases de datos vectoriales son sistemas especializados en el almacenamiento, indexación y recuperación eficiente de vectores de alta dimensión [17][18]. A diferencia de las bases de datos

relacionales, que buscan coincidencias exactas, las bases de datos vectoriales permiten realizar búsquedas por similitud, lo que resulta especialmente útil en aplicaciones de inteligencia artificial y *machine learning* [17][18]. Por ejemplo, en sistemas de recomendación, las bases de datos vectoriales pueden identificar productos o contenidos similares a los que ha consultado o comprado un usuario, mejorando la personalización y la experiencia del cliente [18].

En el ámbito de la IA generativa y los sistemas de recuperación aumentada (RAG), las bases de datos vectoriales son clave para enriquecer las respuestas de los modelos de lenguaje con información contextual y actualizada [18]. Estas bases de datos permiten consultas de baja latencia y escalabilidad, incluso para millones de elementos, lo que las convierte en una herramienta fundamental para aplicaciones modernas de atención al cliente, búsqueda semántica y análisis de datos no estructurados [17][18]. En Colombia, la adopción de estas tecnologías está en crecimiento, impulsando la transformación digital y la innovación en sectores como el financiero, el energético y el gubernamental [19].

Fundamentación Teórica

Teoría de Clasificación de Textos

La clasificación de textos es un pilar fundamental del Procesamiento de Lenguaje Natural (PLN) que consiste en asignar categorías predefinidas a datos no estructurados. Se distinguen tres enfoques principales: el basado en reglas, que utiliza lógica determinista y patrones lingüísticos; el aprendizaje automático (*Machine Learning*), que extrae patrones estadísticos de grandes conjuntos de datos etiquetados; y el enfoque híbrido, que combina la precisión de las reglas expertas con la escalabilidad de los modelos estadísticos para mitigar las debilidades individuales de cada método [62].

SERVQUAL (Calidad del Servicio)

El modelo SERVQUAL, propuesto por Parasuraman, Zeithaml y Berry en el año 1988, constituye un marco teórico esencial para medir la calidad percibida del servicio a través de la brecha entre las expectativas del cliente y su percepción real. Este modelo se estructura en cinco dimensiones críticas: Fiabilidad (cumplimiento de lo prometido), Capacidad de respuesta (disposición a ayudar), Seguridad (conocimiento y confianza), Empatía (atención personalizada) y Elementos tangibles (infraestructura y apariencia) [63].

Sistemas Multi-Agente (MAS)

Wooldridge (2009) define un sistema multi-agente como una sociedad de entidades autónomas capaces de interactuar para resolver problemas que superan las capacidades individuales. Los principios fundamentales de estos sistemas incluyen la autonomía (capacidad de actuar sin intervención directa), la racionalidad (acción orientada a objetivos) y la interacción social (coordinación, negociación y comunicación), permitiendo el modelado de comportamientos

complejos mediante arquitecturas como Creencias-Deseos-Intenciones (BDI) [64].

Gestión de Incidentes (ITIL / ISO 20000)

La gestión de incidentes, bajo los marcos de ITIL e ISO/IEC 20000, se fundamenta en el objetivo primordial de restaurar la operación normal del servicio lo más rápido posible, minimizando el impacto en el negocio. Este proceso teórico sigue un ciclo de vida estructurado que abarca la identificación, registro, categorización, priorización, diagnóstico inicial, escalamiento, resolución y cierre, enfatizando la mejora continua a través de la capitalización del conocimiento generado en cada evento [65].

***Frameworks* de orquestación de agentes inteligentes**

El desarrollo de sistemas basados en agentes inteligentes ha dado lugar a la creación de ecosistemas especializados que facilitan la construcción, orquestación y despliegue de flujos multi-agente sobre diversos LLMs [35]. La selección del *framework* adecuado no es trivial y constituye una decisión arquitectónica de alto impacto en la escalabilidad y mantenibilidad del sistema [34]. A diferencia del encadenamiento secuencial clásico (como las primeras versiones de *LangChain*), las arquitecturas modernas exigen modelar flujos de interacción dinámicos que imiten el razonamiento humano y la colaboración. Entre las alternativas más relevantes del panorama actual se encuentran `LangGraph`, `CrewAI`, `AutoGen`, `Semantic Kernel`, `Haystack y LlamaIndex`, se describen a continuación las alternativas antes mencionadas.

LangGraph: Grafos de Estado para Flujos Agénticos Complejos

`LangGraph` es una biblioteca de Python desarrollada por *LangChain* Inc. e introducida a principios de 2024, diseñada específicamente para construir aplicaciones multi-agente con estado persistente (*stateful*) y flujos de trabajo de larga duración [36]. Su principio arquitectónico central consiste en modelar los flujos de trabajo como grafos dirigidos inspirados en los modelos de computación distribuida de Pregel y Apache Beam, donde cada nodo representa un agente o función, y las aristas controlan el flujo de datos. A diferencia de los *frameworks* lineales, `LangGraph` introduce máquinas de estado que permiten ciclos de retroalimentación, capacidad esencial para que un agente itere sobre sus propias decisiones hasta satisfacer una condición de salida técnica o lógica [37][38][39].

El núcleo operativo de esta herramienta es el *StateGraph*, un objeto que mantiene un estado compartido y persistente (típicamente un *TypedDict*) que cada nodo actualiza de forma secuencial o paralela, garantizando la consistencia de la información sin enrutamientos complejos [37][40]. El sistema incluye mecanismos avanzados como aristas condicionales, subgrafos reutilizables y un proceso de compilación previo a la ejecución que valida la integridad de las conexiones y optimiza las rutas [40]. Con disponibilidad general desde mayo de 2025,

`LangGraph` se ha consolidado en entornos productivos de empresas como Uber y LinkedIn para tareas de detección de amenazas y migraciones de código, apoyándose en la trazabilidad nativa de *LangSmith* para el monitoreo detallado de cada nodo [35][36][41].

***CrewAI*: Orquestación Multi-Agente Basada en Roles**

CrewAI se define como un *framework* de código abierto diseñado para orquestar equipos de agentes mediante una arquitectura basada en roles que emula la estructura colaborativa de equipos humanos [42][43]. El modelo conceptual se organiza en torno a una "crew" o tripulación de agentes especializados que intercambian contexto y artefactos de manera estructurada [44]. Cada entidad se configura mediante tres atributos críticos: el rol (*role*) para su especialización, el objetivo (*goal*) para definir el resultado esperado, y el trasfondo (*backstory*) para dotarlo de un estilo de razonamiento específico. Bajo este esquema, el sistema distingue funciones jerárquicas entre agentes de tipo *Manager*, encargados de la supervisión, y agentes *Worker* o *Researcher*, dedicados a la ejecución y análisis de datos respectivamente [43][45].

La gestión operativa en *CrewAI* se realiza mediante un motor que soporta modelos de ejecución secuencial, paralela y condicional, permitiendo una coordinación jerárquica donde los niveles superiores redistribuyen recursos según la carga de trabajo [43][46]. En términos de capacidades cognitivas, el *framework* proporciona acceso a memoria de corto y largo plazo mediante búferes conversacionales e índices vectoriales, además de soportar arquitecturas RAG para optimizar la recuperación de información externa [42][47]. Actualmente, *CrewAI* es un estándar en el sector empresarial con presencia en gran parte de las empresas Fortune 500, destacando por una arquitectura que prioriza la predictibilidad y la resistencia a comportamientos no deseados frente a modelos de tipo *swarm* [46][48].

***AutoGen*: Conversaciones Multi-Agente y Personalización Dinámica**

Desarrollado por Microsoft *Research*, *AutoGen* es un *framework* pionero en la facilitación de conversaciones autónomas entre agentes capaces de resolver tareas mediante el intercambio de mensajes [56]. Su arquitectura se fundamenta en agentes personalizables y "conversables" que integran LLM, herramientas técnicas y participación humana en un flujo de trabajo fluido. Este ecosistema permite que los agentes alternen roles dinámicamente, como programadores, revisores o ejecutores, según las necesidades del problema, permitiendo que el progreso hacia la meta se dé a través de hilos de diálogo donde los participantes pueden solicitar aclaraciones o corregir errores en tiempo real [56][61].

Una de las ventajas competitivas de *AutoGen* es la integración del concepto *human-in-the-loop*, que permite configurar niveles de supervisión donde un usuario puede aprobar acciones críticas, garantizando que el sistema esté alineado con las políticas de seguridad corporativa [61]. Hacia 2025, el *framework* ha evolucionado para incluir optimización de *prompts* y selección automática de modelos, permitiendo que el sistema asigne la tarea a la IA más eficiente según su complejidad

técnica [56]. Gracias a su capacidad para gestionar estados de conversación extensos y ejecutar código en entornos seguros, la comunidad científica lo reconoce como una solución escalable para la automatización de procesos de investigación y desarrollo de software complejo [60].

***Semantic Kernel*: Integración de Funciones y Planificación Cognitiva**

Semantic Kernel es un SDK de código abierto desarrollado por Microsoft que actúa como un puente entre los lenguajes de programación convencionales y los modelos de lenguaje de gran tamaño [57]. Su arquitectura se aleja del modelo de chat tradicional para centrarse en la integración de "habilidades" (*skills*) y "conectores", tratando las funciones de código como componentes semánticos que el LLM puede invocar. Esta estructura modular, organizada en torno a un núcleo central o *Kernel*, facilita la creación de aplicaciones híbridas donde la lógica de negocio tradicional coexiste con la inteligencia generativa, permitiendo una mantenibilidad excepcional en infraestructuras corporativas robustas [57][61].

El elemento distintivo de este *framework* es su Planificador (*Planner*), un motor cognitivo capaz de desglosar objetivos en lenguaje natural en una serie de pasos lógicos y llamadas a funciones [57]. El Planificador selecciona la secuencia óptima de herramientas para completar una tarea sin necesidad de programación manual de ramificaciones lógicas, transformando la interacción con el software en un modelo basado en intenciones. Además, *Semantic Kernel* implementa una memoria semántica que permite a los agentes recordar interacciones pasadas y acceder a bases de conocimiento externas, posicionándose como una pieza crítica para modernizar aplicaciones legadas mediante el uso de APIs externas de forma segura [57][60][61].

***Haystack*: Tuberías de Procesamiento de Datos y Recuperación Avanzada**

Haystack, desarrollado por *Deepset*, es un *framework* modular orientado a la construcción de aplicaciones de búsqueda semántica y recuperación de datos a gran escala mediante el uso de tuberías (*pipelines*) [58]. En esta arquitectura, la información fluye a través de nodos especializados que realizan tareas granulares, desde el preprocesamiento de documentos hasta la síntesis de respuestas finales. A diferencia de otros orquestadores, Haystack prioriza la implementación de arquitecturas RAG de grado industrial, integrándose con una amplia variedad de bases de datos vectoriales y motores de búsqueda tradicionales para asegurar una precisión técnica superior en el manejo de datos documentales [58][60].

La evolución reciente del sistema ha incorporado generadores y enrutadores que permiten una lógica de control dinámica, mitigando el riesgo de alucinaciones mediante la validación de hechos en tiempo real [58]. Este enfoque pragmático en la integridad de la información lo hace especialmente apto para sectores regulados, como el legal o el financiero, donde la trazabilidad de la fuente es obligatoria. Su diseño transparente, reforzado por una comunidad que impulsa componentes para el *re-rankeo* y expansión de consultas, permite monitorear individualmente cada entrada y salida del flujo, consolidándolo como el referente para proyectos donde la eficiencia en la recuperación de datos es el pilar fundamental [58][60].

LlamaIndex: Orquestación de Datos y Agentes Basada en Conocimiento

LlamaIndex es un *framework* especializado en la conexión de datos privados con modelos de lenguaje, evolucionando desde la indexación hacia una plataforma completa de orquestación agéntica [59]. Su premisa central es que la inteligencia de un agente está condicionada por la accesibilidad de los datos en su contexto; por ello, ofrece herramientas para ingestar y estructurar información desde fuentes heterogéneas, creando una capa de abstracción navegable. En este ecosistema, los agentes actúan como razonadores sobre índices de datos complejos mediante un motor de consulta (*Query Engine*), decidiendo la estrategia de recuperación más adecuada, ya sea analizando resúmenes de alto nivel o profundizando en detalles técnicos específicos según se requiera [59][60].

El sistema cuenta con una suite de *Data Agents* diseñados para realizar tareas autónomas de lectura y escritura sobre repositorios de información, generando informes técnicos o análisis sobre tablas estructuradas [59]. Hacia 2025, *LlamaIndex* ha fortalecido su capacidad de orquestación mediante flujos de trabajo basados en eventos, permitiendo definir procesos de negocio complejos con un control de estados similar al de los grafos de computación. Esta evolución garantiza que el sistema no solo sea eficiente recuperando datos, sino también ejecutando acciones corporativas que dependen de la veracidad de la información almacenada, integrándose perfectamente con ecosistemas de observabilidad para cerrar la brecha entre datos estáticos y agentes dinámicos [59][60].

Referentes y Aplicaciones en Colombia

La implementación de la inteligencia artificial (IA) y las metodologías de diseño centrado en el usuario han experimentado un crecimiento sostenido en Colombia, impulsadas por una inversión empresarial ascendente y el surgimiento de un ecosistema de *startups* altamente innovadoras [19][20]. Este avance se fundamenta en una robusta hoja de ruta nacional, liderada por la **Política Nacional de Inteligencia Artificial (CONPES 4144)**, la cual establece los lineamientos estratégicos para garantizar una adopción ética y productiva de estas tecnologías tanto en el sector público como en el privado [22].

En el ámbito de los servicios públicos, **Empresas Públicas de Medellín (EPM)** ha destacado como referente en la integración de sistemas inteligentes para la gestión optimizada de incidentes. Mediante la implementación de agentes virtuales que analizan datos de consumo y cobranza en tiempo real, la entidad ha logrado gestionar más de dos millones de interacciones, optimizando la atención al cliente en el sector eléctrico y demostrando la capacidad de la IA para manejar procesos críticos de consulta y seguimiento de servicios [19][21].

Por su parte, el sector financiero ha adoptado la transformación digital mediante el uso de biometría avanzada y reconocimiento facial, con empresas como **DiBanka y Bytte** liderando la agilización de procesos de verificación. Estas soluciones tecnológicas han permitido reducir

significativamente los tiempos de procesamiento de préstamos, apoyándose en modelos predictivos que garantizan mayor seguridad y eficiencia en la prestación de servicios financieros tradicionales y digitales [19][20].

La versatilidad de la IA se extiende también al sector salud y al comercio minorista, donde se aplican algoritmos avanzados para mejorar la calidad de vida y la experiencia del consumidor. Mientras la **Fundación Santa Fe de Bogotá** utiliza modelos de visión artificial para acelerar diagnósticos médicos mediante la mejora de imágenes, el **Grupo Éxito** emplea analítica predictiva para ofrecer descuentos personalizados, evidenciando cómo la IA permite una atención más precisa y una relación más estrecha con el usuario [20].

A pesar de este dinamismo, el panorama actual enfrenta desafíos estructurales que condicionan la velocidad de su despliegue, tales como las limitaciones en la infraestructura digital y la necesidad de un marco regulatorio aún más robusto [19][20]. Asimismo, la escasez de talento humano capacitado sigue siendo una barrera crítica, lo que subraya la urgencia de fortalecer la colaboración entre el gobierno, la industria y la academia para consolidar un ecosistema tecnológico sólido y competitivo a nivel regional [19].

En respuesta a estas necesidades, la investigación académica en Colombia, documentada en repositorios de prestigio como la **ACL Anthology** y la **SEPLN**, ha realizado aportes fundamentales en el **Procesamiento de Lenguaje Natural (PLN)** aplicado al español latinoamericano. Estos estudios abordan retos específicos relacionados con regionalismos y contextos socioculturales, proporcionando la base científica necesaria para que las herramientas de IA, como los agentes de clasificación de reportes, sean técnicamente robustas y contextualmente pertinentes [20][65].

Finalmente, la maduración de este ecosistema se hace evidente en el éxito de firmas como **Nuvu**, reconocida globalmente como socio estratégico de servicios en la nube, y en la adopción institucional por parte de la Policía Nacional y administraciones locales como la Alcaldía de Cali [19]. Estas iniciativas, junto con el esfuerzo académico, reflejan un compromiso país por cerrar brechas de productividad y mejorar la provisión de servicios ciudadanos, posicionando a Colombia como un actor relevante en la transformación digital de la región [64][65].

2.3.2 ANTECEDENTES

La gestión eficiente de reportes de incidentes en el sector eléctrico es un desafío ampliamente documentado, especialmente cuando se busca optimizar la experiencia del cliente, reducir costos operativos y mejorar la asignación de recursos técnicos. Para comprender la pertinencia y el alcance de las tecnologías empleadas en este proyecto, resulta necesario recorrer la trayectoria histórica de los avances en inteligencia artificial, procesamiento de lenguaje natural, bases de datos vectoriales y metodologías de diseño centrado en el usuario que le dan sustento.

La historia de la inteligencia artificial comienza mucho antes de que el término existiera

formalmente. En 1950, el matemático británico Alan Turing publicó su influyente artículo *Computing Machinery* and *Intelligence*, en el que planteó la pregunta "¿pueden las máquinas pensar?" y propuso lo que hoy se conoce como el Test de Turing, que es un experimento en el cual un interrogador humano debe determinar, mediante una conversación, si su interlocutor es una persona o es una máquina. En lugar de definir filosóficamente la inteligencia, Turing propuso evaluarla a partir de un comportamiento observable, un enfoque pragmático y revolucionario para la época que sentó las bases conceptuales de todo lo que hoy llamamos inteligencia artificial [24][66].

Seis años después, en el verano del año 1956, el informático John McCarthy organizó una histórica Conferencia de Dartmouth junto a Marvin Minsky, Claude Shannon y Nathaniel Rochester. Fue allí donde McCarthy emitió formalmente el término conocido como inteligencia artificial, y la definió como "la ciencia e ingeniería de hacer máquinas inteligentes", y donde el campo quedó establecido como disciplina académica independiente. Aquel evento se considera hasta hoy el punto de partida oficial de la IA como área de investigación [24].

Durante las décadas siguientes, la IA avanzó a través de ciclos de entusiasmo y escepticismo, los llamados "inviernos de la IA", hasta que, en 1986, Geoffrey Hinton junto a David Rumelhart y Ronald Williams desarrollaron y popularizaron el algoritmo de retropropagación (*backpropagation*), que permitió entrenar redes neuronales multicapa de manera eficiente y abrió el camino hacia arquitecturas mucho más profundas. Dos décadas más tarde, en el año 2006, el propio Hinton publicó su trabajo seminal sobre *Deep Belief Networks*, demostrando que redes de múltiples capas podían entrenarse de forma efectiva capa por capa. Ese artículo reencendió el interés global en las redes neuronales y dio el inicio a lo que hoy en día conocemos como la era del aprendizaje profundo [25].

El año 2011 marcó otro hito de gran relevancia pública, IBM Watson venció a dos campeones humanos en el concurso televisivo Jeopardy, demostrando así, ante el mundo el potencial tan grande de la IA y del procesamiento de lenguaje natural para comprender y responder a preguntas complejas formuladas en nuestro lenguaje humano. Un año después, en el año 2012, el equipo de Google Brain, encabezado por Jeff Dean y Andrew Ng, desarrollaron redes neuronales profundas capaces de detectar patrones en imágenes y videos sin alguna supervisión explícita, marcando así, el inicio de la explosión masiva de aplicaciones de aprendizaje profundo en la industria. En el ámbito empresarial, estas capacidades se tradujeron rápidamente en soluciones concretas de atención al cliente. IBM Watson fue implementado en distintos bancos y aseguradoras para automatizar la gestión de consultas y poder analizar grandes volúmenes de datos textuales, permitiendo validar tempranamente la viabilidad del uso de IA en operaciones de servicio a gran escala [25][27].

En paralelo a estos avances de aprendizaje automático, el procesamiento de lenguaje natural también recorrió su propio camino. Entre los años 2010 y 2013, Tomas Mikolov y su equipo en Google desarrollaron Word2Vec, que consistió en una técnica que permitió representar palabras

como vectores numéricos en espacios de alta dimensión, de modo que términos con significados similares quedaran ubicados cerca entre sí. Esta capacidad de capturar relaciones semánticas fue fundamental para tareas como la clasificación de textos y el análisis de intenciones en lenguaje natural [33].

Entre los años 2012 y 2014, los modelos de aprendizaje profundo aplicados al NLP comenzaron a superar sistemáticamente a los métodos basados en reglas de tareas como el reconocimiento de entidades y la clasificación de intenciones. El salto definitivo llegó en 2017, cuando investigadores de Google publicaron el artículo "*Attention Is All You Need*", introduciendo la arquitectura *Transformer*, que transformó radicalmente el campo. Sobre esta arquitectura se construyeron posteriormente modelos como BERT y la familia GPT, a partir del año 2018, con los cuales se demostró una capacidad sin precedentes para comprender y generar texto humano, revolucionando así, la interacción con sistemas automatizados en prácticamente todos los sectores productivos [33][24].

La aplicabilidad de estas tecnologías al problema específico de clasificación de reportes de incidentes fue validada en trabajos como el desarrollado por BioAITeam para el área de soporte técnico de una empresa de software, donde la combinación de ML y NLP logró automatizar la categorización de solicitudes [28]. No obstante, la aplicación más directa al sector de *utilities* eléctricas se encuentra en investigaciones publicadas en IEEE y **Electric Power Systems Research**. Por ejemplo, se ha propuesto el uso de modelos basados en BERT *fine-tuned* con terminología técnica para la clasificación automática de reportes de fallas en redes de distribución, logrando identificar causas raíz con alta precisión. De igual forma, en el ámbito de la seguridad informática, Baciero Fernández documentó el uso de sistemas multiagente para la priorización automática de incidentes, demostrando que los agentes especializados pueden gestionar flujos complejos de reportes en tiempo real [31][67][68].

A medida que el volumen de datos no estructurados creció exponencialmente durante la década de 2010, surgió la necesidad de sistemas con capacidad de ir más allá de las búsquedas por coincidencia exacta. Las bases de datos vectoriales respondieron a esta necesidad: sistemas especializados en almacenar y consultar las representaciones numéricas que los modelos de lenguaje generan para cada fragmento de texto, permitiendo búsquedas por similitud semántica sobre millones de registros con baja latencia. En la década de 2020, con el auge de la IA generativa, estas bases de datos se convirtieron en componente central de los sistemas de Recuperación Aumentada por Generación (RAG), que permiten enriquecer las respuestas de los modelos de lenguaje con información contextual actualizada. Empresas como Microsoft, AWS y Meta lanzaron productos comerciales sobre esta tecnología, habilitando así, la implementación de asistentes virtuales y sistemas de búsqueda semántica en entornos de producción a nivel empresarial [17][18][30].

En cuanto a las metodologías de diseño centrado en el usuario, sus fundamentos se remontan al libro Las Ciencias de lo Artificial, publicado en 1969 por Herbert Simon, Premio Nobel de

Economía, donde se introdujo por primera vez el concepto de diseño como proceso cognitivo orientado a la resolución de problemas. Durante los años setenta, la Universidad de Stanford comenzó a desarrollar estas ideas de forma teórica, y en 1991 David Kelley fundó la consultoría IDEO en Palo Alto, California, llevando el enfoque al ámbito empresarial con un modelo radicalmente colaborativo y multidisciplinario. La metodología adquirió su nombre definitivo y su estructura de cinco fases, empatizar, definir, idear, prototipar y testear en el año 2008, cuando Tim Brown, CEO de IDEO, publicó el artículo "*Design Thinking*" en la Harvard Business Review. Desde entonces, empresas líderes del mercado, como lo son SAP y Airbnb la aplicaron para rediseñar productos y servicios desde la perspectiva del usuario final, obteniendo de esta forma, mejoras significativas tanto en la experiencia del cliente como en la eficiencia operativa [11][26][29].

A partir de 2016, la inteligencia artificial se convirtió en un eje central para la optimización de la experiencia del cliente en servicios públicos y otros sectores. Los chatbots y asistentes virtuales comenzaron a desplegarse masivamente, ofreciendo atención disponible las 24 horas del día los 7 días de la semana y aprendiendo de las interacciones para mejorar su desempeño de forma continua [32][27].

En el contexto latinoamericano, el antecedente más directo y relevante para este proyecto es el de Empresas Públicas de Medellín (EPM), la empresa de servicios públicos más grande de Colombia. En junio de 2019, EPM lanzó a EMA, su asistente virtual que está basada en inteligencia artificial y NLP, y está disponible a través del sitio web corporativo y de WhatsApp para el reporte de fallas, consulta de facturas y otras transacciones de autogestión del cliente. Para mayo del año 2024, EMA ya había acumulado más de 5,2 millones de interacciones con la comunidad, y el 26% de todas las interacciones digitales de la empresa se canalizaban por este medio, lo que representa un avance sobre el estado del arte documentado en *utilities* latinoamericanas [53][54][55].

Sin embargo, existe una diferenciación fundamental con la propuesta de Celsia: mientras que EMA opera como un asistente virtual generalista enfocado en la omnicanalidad y el servicio al cliente transaccional, el proyecto desarrollado para Celsia se centra específicamente en la capa de **inteligencia operativa para la clasificación automática de reportes técnicos**. A diferencia de EMA, que actúa principalmente como interfaz de recepción, esta propuesta implementa un motor analítico que categoriza automáticamente el contenido técnico de los incidentes para su integración directa en el Sistema de Gestión de interrupciones (OMS), optimizando el triaje y la asignación de recursos técnicos de manera autónoma [53][54].

Esta trayectoria, que inicia desde la pregunta de Turing sobre si las máquinas pueden pensar, hasta hoy en día, los asistentes virtuales desplegados en empresas de servicios públicos latinoamericanas, demuestra que la evolución de la IA, el NLP, las bases de datos vectoriales y las metodologías de diseño han convertido en herramientas ya maduras y probadas para abordar exactamente el tipo de problema que enfrenta Celsia. La propuesta de este proyecto no parte de

cero: se apoya en décadas de avances científicos e industriales validados, adaptándolos al contexto operativo específico del sector eléctrico en los departamentos del Valle del Cauca y Tolima.

3. DISEÑO DE FLUJO CONVERSACIONAL CENTRADO EN EL CLIENTE

Este capítulo presenta los resultados del primer objetivo específico del proyecto, diseñar un flujo de interacción centrado en el cliente, utilizando técnicas de *Design Thinking*, que optimice la recolección de información y facilite la categorización inicial de los reportes de incidentes según el tipo de daño. El entregable principal de este objetivo es el diagrama del flujo conversacional propuesto (*TO-BE*), que se presenta en la sección 3.4.

Para llegar a ese entregable, el capítulo recorre tres caminos complementarios: primero, describe el contexto operativo y la evolución histórica del ecosistema digital de Celsia, que configura el escenario sobre el cual opera el problema; segundo, documenta el proceso de *Design Thinking* llevado a cabo con especialistas del equipo de Experiencia al Cliente y del equipo de Tecnología de Celsia, cuyas sesiones de levantamiento constituyeron la fuente primaria de información para el diseño; y tercero, analiza la evolución del flujo conversacional desde su versión original de 2022 hasta su estado actual, para fundamentar con evidencia las decisiones de diseño del flujo propuesto.

3.1 Contexto operativo: Evolución del ecosistema digital en Celsia

Para comprender el problema en su dimensión real, resulta fundamental describir la trayectoria de transformación digital que Celsia ha experimentado en su canal de atención LuzIA desde el año 2022. Esta evolución configura el escenario en el cual opera el problema que este proyecto busca resolver y explica por qué la solución propuesta no parte de cero, sino que se construye sobre una infraestructura ya existente.

Hasta septiembre de 2022, la atención de reportes de daños en Celsia se gestionaba principalmente a través del canal telefónico. Con la introducción del sistema LuzIA Voz, un asistente virtual basado en IA conversacional, en apenas un mes fue posible trasladar el 13% de las interacciones del canal telefónico hacia canales digitales sin intervención de un asesor humano. Sin embargo, esta transición inicial reveló un efecto secundario inesperado, que fue el incremento de quejas por parte de los clientes, probablemente asociado a la falta de validaciones en las primeras versiones del flujo digital y a que los clientes tenían ahora más oportunidades de interacción sin recibir respuestas satisfactorias.

A partir de abril de 2023, Celsia incorporó WhatsApp como canal de atención masivo, lo que amplió significativamente el alcance del ecosistema LuzIA y permitió que los clientes encontraran un valor real en el canal digital para reportar fallas en el servicio. En sus inicios, el flujo de WhatsApp replicaba lo mismo del portal web sin ningún tipo de validación semántica, lo que ocasionó que las brigadas técnicas fueran despachadas directamente a sitio basándose en reportes que, al llegar, no correspondían a fallas reales de la red eléctrica: el cliente había reportado un problema de internet, una inconformidad con la facturación o una solicitud de reconexión, entre otros casos. Estas visitas fallidas consumían tiempo y capacidad operativa de

los equipos en campo y generaban costos directos sin haber resuelto ningún incidente real.

Fue a partir de ese aprendizaje que Celsia implementó un equipo humano dedicado exclusivamente a revisar cada reporte recibido y determinar si correspondía a una falla del servicio de energía o si debía dirigirse a otra área antes de escalar el caso a las brigadas. Esta clasificación manual, aunque efectiva como filtro, representaba una solución de alto costo operativo y baja escalabilidad frente al volumen creciente de interacciones digitales.

Hacia el año 2025, el volumen de interacciones había crecido hasta un punto en que mantener ese equipo de personas dejó de ser sostenible. En conjunto con el equipo de Tecnología, se realizaron ajustes al flujo conversacional de WhatsApp para incorporar un primer filtro automático basado en inteligencia artificial generativa, que redujo la dependencia de la revisión manual y elevó la efectividad de clasificación de un 40% a un 95% aproximadamente. Sin embargo, esta mejora dejó por fuera del alcance una serie de tipologías que aún hoy en día, continúan ingresando por el flujo de energía sin ser correctamente redirigidas: reportes de aseo, alumbrado público, reclamaciones, clientes en zonas de peaje y solicitudes de reconexión siguen mezclados con los incidentes operativos reales, generando ruido en el sistema y manteniendo una proporción de conversaciones no clasificables que el ajuste de 2025 no logró eliminar. Es precisamente en esa brecha donde se sitúa la propuesta de este proyecto.

En el año 2026, el ecosistema LuzIA, que integra los canales de voz y WhatsApp, gestiona el 93% de los reportes por medios digitales sin intervención de asesor humano, reservando la atención personalizada para casos especiales como lo son los clientes del sector A, las reconexiones y los clientes en zonas de peaje (clientes frontera). Al corte del 12 de marzo de 2026, el canal de WhatsApp acumulaba más de 1.004.292 conversaciones, con un promedio mensual de 83.707 interacciones. De estas, el 19,6% correspondía a gestión de daños y el 9,6% permanecía sin clasificar correctamente, lo que evidencia la oportunidad de mejora que este proyecto aborda, casi una de cada diez conversaciones no puede procesarse operativamente por falta de categorización adecuada.

La Tabla 1 sintetiza la evolución del ecosistema LuzIA desde su implementación, mostrando para cada período el hito tecnológico alcanzado, las métricas asociadas y la limitación que motivó la siguiente mejora.

Tabla 1. Evolución del ecosistema LuzIA - métricas por período (2022–2026).

Período	Hito	Métricas clave	Tecnología / Canal	Limitación revelada
Antes de sep. 2022	Atención telefónica sin canal digital	100% de reportes por voz	Call center + operadores humanos	Alto costo, baja escalabilidad
Sep.	Lanzamiento	13% de	Asistente	Incremento de quejas

2022	LuzIA Voz	interacciones migradas a digital en el primer mes	virtual IA conversacional	por falta de validaciones en el flujo
Abr. 2023	Incorporación de WhatsApp como canal masivo	Visitas fallidas sin cuantificar; brigadas despachadas sin filtro semántico	WhatsApp replicando flujo del portal web (sin NLP)	Reportes de internet, facturación y reconexión mezclados con energía
2025	Migración CLU → LLM (Azure OpenAI + Twilio)	Efectividad de clasificación: ~40% → ~95% 9,6% residual sin clasificar	LLM clasificador + RAG + agente de facturas	Tipologías no-energéticas siguen mezcladas; 9,6% sin resolver
Mar. 2026 (corte)	Estado actual del ecosistema LuzIA	>1.004.292 conversaciones acumuladas 83.707 interacciones/mes 93% digital sin operador 9,6% sin clasificar	LuzIA Voz + WhatsApp integrados	Brecha residual del 9,6% y tipologías ajenas aún en flujo de energía

Los municipios con mayor concentración de reportes son Ibagué (14,08%), Espinal (5,95%), Palmira (5,58%) y Melgar (4,23%), correspondientes a las zonas operativas de Tolima y Valle del Cauca. La distribución geográfica no es homogénea: existen zonas con alta densidad de reportes que concentran la mayor presión sobre las brigadas técnicas y donde el impacto de una clasificación incorrecta es operativamente más costoso. Es sobre esta realidad operativa que se diseña la solución propuesta en los capítulos siguientes.

3.2 Metodología *Design thinking* aplicada

La propuesta de rediseño del flujo conversacional se desarrolló siguiendo las cinco fases del *Design Thinking*, adaptadas al contexto operativo de Celsia. La elección de esta metodología responde a la necesidad de abordar el problema desde una perspectiva centrada en el usuario, reconociendo que el origen de las clasificaciones erróneas no es exclusivamente tecnológico, sino también de diseño del proceso y de la interacción humano-sistema.

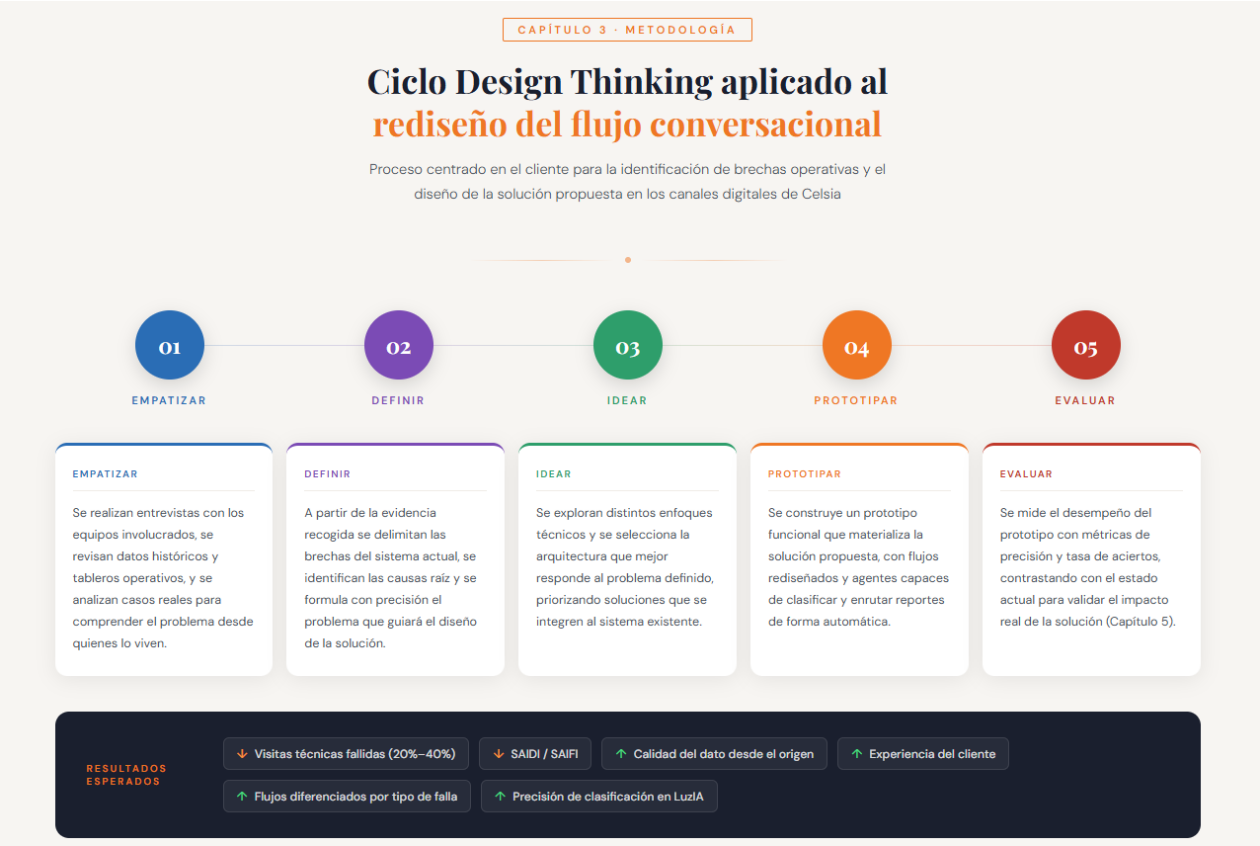


Ilustración 2. Ciclo Design Thinking aplicado al rediseño del flujo conversacional de reportes de daños. Fuente: Elaboración propia con apoyo de IA Claude a partir del diseño conceptual del proyecto.

3.2.1 Fase 1 - Empatizar

Con el objetivo de comprender el problema desde la experiencia real de quienes interactúan con el sistema, se realizaron dos sesiones de trabajo con cada uno de los tres perfiles de especialistas de Celsia descritos a continuación. Las sesiones combinaron entrevistas semiestructuradas, revisión de tableros de control operativos en Power BI, análisis del *dataset* histórico de más de un millón de registros de reportes provenientes de ADMS, y revisión de los *prompts* y configuraciones actuales del ecosistema LuzIA.

3.2.2 Criterio de selección de participantes

Los participantes fueron seleccionados mediante un criterio de selección intencional por rol estratégico, priorizando perfiles con conocimiento directo y complementario del sistema LuzIA en sus distintas dimensiones. Se establecieron tres criterios de inclusión:

- Participación en el diseño, operación o mantenimiento del ecosistema LuzIA durante al menos un año.

- Acceso a información operativa de primera línea sobre el flujo de clasificación de incidentes
- Representación de las tres perspectivas clave del sistema: experiencia del cliente, arquitectura tecnológica y operación en campo. Se excluyeron perfiles con conocimiento únicamente administrativo o comercial, al no tener incidencia directa sobre el flujo de clasificación objeto de estudio.

Protocolo de sesiones

Se realizaron cinco sesiones en total, con una duración variable entre una y una hora y media según el participante y la profundidad temática de cada encuentro. La primera sesión con cada participante tuvo un carácter exploratorio, orientada a comprender su rol, la evolución del sistema y los principales puntos de fricción identificados en su experiencia. La segunda sesión, realizada únicamente con Sebastián, fue de profundización, centrada en la revisión de casos concretos, datos operativos y validación de los hallazgos preliminares. Todas las sesiones fueron realizadas de manera virtual mediante Microsoft Teams. Los participantes firmaron un consentimiento informado antes de iniciar la primera sesión (ver Anexo 1, Anexo 2 y Anexo 3). La guía de preguntas utilizada en cada sesión se encuentra disponible en el Anexo 4

Anexo 1. Consentimiento informado Luis Carvajal.docx
Anexo 2. Consentimiento informado Sebastián Gómez.docx
Anexo 3. Consentimiento informado Darío Díaz.docx
Anexo 4. Guía de entrevista semiestructurada.docx

Tabla 2. Registro de sesiones de levantamiento de información — Fase de Empatía.

Participante	Perfil	Sesión	Fecha	Duración
Luis Carvajal	Experiencia al Cliente	Sesión 1	13/03/2026	1,5 horas
Sebastián Gómez	Tecnología	Sesión 1	16/02/2026	1 hora
Sebastián Gómez	Tecnología	Sesión 2	27/03/2026	1 hora
Darío Díaz Gómez	Operación	Sesión 1	09/06/2025	1,5 horas

En este contexto, y con el fin de estructurar de manera clara la información recolectada durante las sesiones de trabajo, se caracterizaron los tres perfiles clave que interactúan con el sistema LuzIA desde diferentes perspectivas: experiencia del cliente, tecnología y operación. A continuación, se presentan los participantes involucrados en la fase de empatía, detallando su rol dentro del proceso, su experiencia en el ecosistema y la metodología aplicada en cada sesión.

Tabla 3. Participante 1 - Experiencia al Cliente

Luis Carvajal · Especialista de Experiencia al Cliente	
Rol en el proceso	Diseño del flujo conversacional, integración de sistemas (ADMS, WFM, API multicanal), gestión de canales digitales: WhatsApp, Portal Web, App Clientes y LuzIA Voz.
Experiencia / perfil	Participación directa desde las fases de diseño, prototipado, pruebas (4 meses de piloto controlado) y escalamiento del ecosistema LuzIA desde su nacimiento en 2022 hasta su estado actual en 2026.
Metodología de sesión	Entrevista semiestructurada. Revisión en vivo de tableros Power BI operativos, análisis cronológico de la evolución del sistema y discusión de casos representativos de fallas en la clasificación.

Tabla 4. Participante 2 - Tecnología (TI)

Sebastián Gomez Roldan · Especialista de Tecnología	
Rol en el proceso	Arquitectura técnica del ecosistema LuzIA: integración con Twilio (WhatsApp), Azure OpenAI (LLM), Azure Cognitive Services (CLU), App Services, Blob Storage y sistemas comerciales.
Experiencia / perfil	Conocimiento end-to-end de la infraestructura desde la versión CLU original (2022) hasta el sistema LLM + RAG actual (2025). Responsable de la integración entre LuzIA y los sistemas ADMS y WFM.
Metodología de sesión	Entrevista técnica semiestructurada. Revisión de arquitectura de sistemas, walkthrough del flujo de integración actual y compartición de los prompts operativos del sistema clasificador.

Tabla 5. Participante 3 - Operación (Centro de Control)

Darío Díaz Gómez · Especialista de Operación - Centro de control	
Rol en el proceso	Supervisión y seguimiento en tiempo real de incidentes registrados en ADMS y coordinación de la respuesta operativa en campo para los departamentos del Valle del Cauca y Tolima.
Experiencia / perfil	Experiencia directa en la gestión de brigadas, análisis de reportes de daños y detección de visitas fallidas. Conocimiento de los costos operativos asociados al despacho innecesario y del impacto en los indicadores SAIDI y SAIFI.
Metodología de sesión	Entrevista semiestructurada centrada en el flujo operativo post-clasificación: desde que el incidente se crea en ADMS hasta que la brigada cierra o reporta una visita fallida. Revisión de casos representativos del Centro de Control.

A partir de la información obtenida en las entrevistas y el análisis conjunto de los tres perfiles, se consolidaron los principales hallazgos de la fase de empatía. Estos resultados se agrupan en tres bloques: diseño del ecosistema, limitaciones del sistema actual y oportunidades de mejora.

La siguiente tabla resume de manera estructurada estos hallazgos, integrando las perspectivas de las áreas de Experiencia del Cliente, Tecnología y Operación.

Tabla 6. Perfiles de participantes y hallazgos consolidados de la fase de empatía - Design Thinking. Fuente: Elaboración propia

Hallazgos consolidados de la fase de empatía	
Bloque **1 — Diseño y construcción del ecosistema LuzIA**	
Principio de diseño	El equipo mapeó la experiencia del cliente negociando qué campos del formulario serían obligatorios y cuáles opcionales. TI estuvo involucrado desde el inicio para garantizar viabilidad técnica. Siempre se diseñó con enfoque multicanal y anticipando todos los casos de uso posibles antes de prototipar.
Cultura de mejora continua	«Lo perfecto es enemigo de lo bueno»: Celsia prioriza pilotos controlados, pruebas iterativas y aprendizaje continuo sobre soluciones perfectas desplegadas en una sola etapa. Este principio guió cada iteración del flujo conversacional.
Transición digital 2022	En un solo mes (sep.-oct. 2022) se redujo el 13% de las llamadas telefónicas. Las quejas aumentaron inicialmente por falta de validaciones, lo que impulsó las mejoras posteriores. A partir de 2023, con WhatsApp, los clientes comenzaron a encontrar valor genuino en el canal digital.
Bloque **2 — Limitaciones del sistema actual (2025–2026)**	
Brecha no clasificable	El 9,6% de las conversaciones de WhatsApp permanece sin clasificar al corte del 12 de marzo de 2026. El sistema actual responde a estos casos con un mensaje genérico de disculpa y remite al *call center*, sin extraer ningún dato operativo útil.
Tipologías sin filtro	Reportes de aseo, alumbrado público, internet, reclamaciones, reconexiones y clientes en zonas de peaje continúan ingresando al flujo operativo de energía sin ser redirigidos, generando incidentes mal clasificados en ADMS.
Visitas fallidas y costos	Cuando una clasificación errónea llega a ADMS, el Centro de Control despacha una brigada que, al llegar al sitio, encuentra que no existe falla real en la red. Esto genera costos directos (combustible, horas-hombre, desgaste vehicular) y costos indirectos (replanificación, carga administrativa), además de deteriorar los indicadores SAIDI y SAIFI al retrasar la atención de incidentes genuinos. (Hallazgo: Darío Díaz Gómez, Centro de Control)

Restricción Meta / Twilio	WhatsApp exige que todas las respuestas se entreguen en menos de 10 segundos. Si el sistema se demora más, la conversación se cae. Esto limita la profundidad del procesamiento semántico en tiempo real y condiciona la arquitectura de los agentes propuestos.
Bloque **3 — Oportunidades identificadas**	
Base LLM existente	La infraestructura de LLM implementada en 2025 (Azure OpenAI + Twilio) ofrece una base sólida sobre la cual agregar agentes especializados de mayor precisión semántica sin necesidad de reemplazar la arquitectura actual.
Arquitectura de respuesta	La restricción de 10 segundos define un patrón de diseño: clasificador inicial rápido (modelo básico) + agentes de profundidad semántica ejecutándose en segundo plano con retoma de conversación, compatible con las ventanas de 24 horas de WhatsApp.
Gestión proactiva de expectativas	Informar al cliente el tiempo estimado de atención según zona y tipo de falla es un diferenciador clave identificado por el equipo de Experiencia al Cliente. Cada dato entregado al cliente genera expectativa de más información: el sistema debe diseñarse con esta dinámica en mente.
Data set histórico disponible	Más de un millón de registros de reportes históricos en ADMS, con texto libre del cliente y clasificación final asignada, constituyen el corpus de entrenamiento y validación para los *embeddings* y el modelo de clasificación propuesto en el Capítulo 4.

3.2.3 Fase 2 - Definir

A partir de los hallazgos de la fase de empatía, se definió que el problema no es la ausencia total de clasificación automática, pues desde agosto de 2025 el ecosistema LuzIA ya incorpora LLMs y un sistema de agentes que apoya la categorización, sino la persistencia de una brecha residual que ese sistema no ha logrado cerrar. Los datos al corte del 12 de marzo de 2026 evidencian que el 9,6% de las conversaciones de WhatsApp aún permanece sin clasificar, y que tipologías como aseo, alumbrado público, reclamaciones, reconexiones y clientes en zonas de peaje continúan ingresando al flujo de energía sin ser correctamente redirigidas.

Para sintetizar los hallazgos de la fase de empatía y formular el problema desde la perspectiva del usuario, se construyó un enunciado POV (*Point of View*) que integra las tres perspectivas recogidas:

"Luis Carvajal, Sebastián y Darío Díaz Gómez, como especialistas de Experiencia al Cliente, Tecnología y Operación de Celsia, necesitan un mecanismo que discrimine con precisión el tipo de solicitud del cliente desde el primer mensaje, porque el sistema actual clasifica correctamente el 90,4% de las conversaciones, pero deja sin resolver un 9,6% que genera despachos innecesarios, costos operativos evitables y deterioro de los indicadores SAIDI y SAIFI."

Complementariamente, se construyó un mapa de experiencia del cliente que recorre el flujo actual desde que el usuario envía su mensaje por WhatsApp hasta que la brigada cierra o reporta una visita fallida (Ver Ilustración 3) Este mapa identificó dos momentos críticos de fricción: el momento en que el cliente describe libremente su problema sin recibir preguntas de diagnóstico, lo que impide capturar información clave sobre la naturaleza real del incidente; y el momento en que el sistema asigna una categoría sin verificar la coherencia con la descripción textual del cliente, lo que deriva en que el 9,6% de las conversaciones quede sin clasificar y en que tipologías ajenas al servicio eléctrico continúen ingresando al flujo de energía.

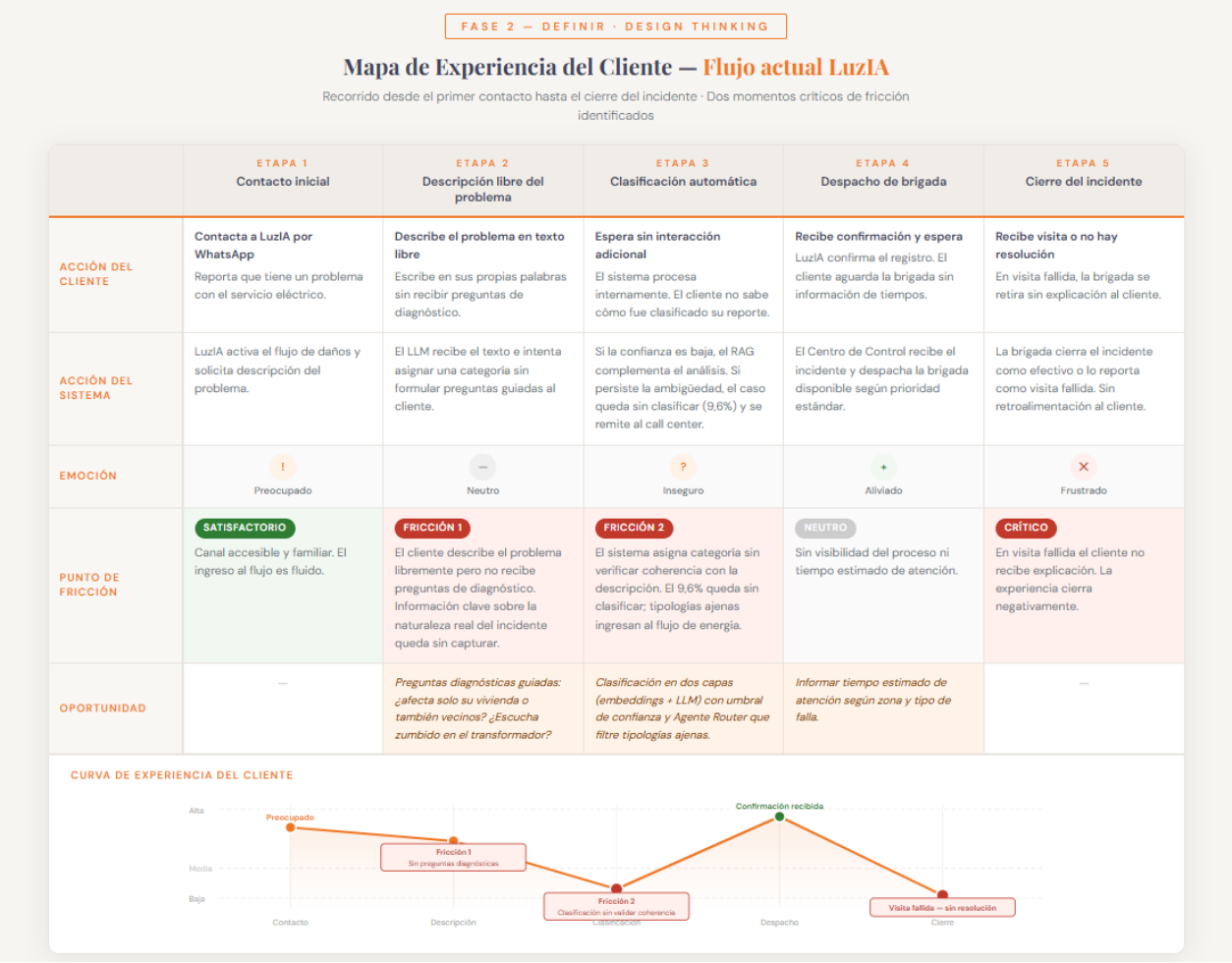


Ilustración 3. Mapa de experiencia al cliente- Flujo AS-IS. Fuente: Elaboración propia. Diagramación asistida por Claude.

El problema central se define como la insuficiencia del flujo conversacional actual para discriminar con precisión entre incidentes reales del servicio de energía y reportes que corresponden a otras áreas, lo que se traduce en tres consecuencias operativas concretas: creación de incidentes mal clasificados en ADMS que no corresponden a fallas reales del servicio; despacho de brigadas a sitios sin daño de red, con el consecuente impacto en costos operativos y en los indicadores SAIDI

y SAIFI; y saturación del flujo operativo de energía con reportes de servicios no relacionados que el sistema actual no filtra de manera exhaustiva.

3.2.4 Fase 3 - Idear

La fase de ideación tomó como punto de partida el estado actual del sistema en 2025, reconociendo que la solución debe construirse sobre la infraestructura de LLM ya existente en LuzIA, no reemplazarla. Para seleccionar el enfoque técnico más adecuado, el equipo evaluó tres alternativas de diseño, analizando las fortalezas y limitaciones de cada una frente a las dos fricciones identificadas en la fase de definición. La Tabla 7 sintetiza este proceso de evaluación.

Tabla 7. Alternativas técnicas evaluadas en la fase de ideación.

Enfoque evaluado	Descripción	Fortalezas	Limitaciones identificadas	Decisión
RAG con preguntas diagnósticas por cliente	Formular preguntas específicas al cliente mediante RAG antes de clasificar el reporte.	Alta personalización. Aprovecha contexto e historial del cliente.	Flujo extenso que afecta la experiencia en mensajería inmediata. Ineficiente como clasificador principal dado el volumen de interacciones.	**Parcialmente incorporado**
Clasificación solo por similitud vectorial (embeddings)	Clasificar el texto del cliente por similitud vectorial contra el dataset histórico de ADMS, sin LLM.	Rápido, de bajo costo computacional y basado en historial operativo real.	No captura matices semánticos complejos. Desaprovecha la capacidad LLM ya instalada desde 2025.	**Descartado como solución única**
Embeddings + LLM con agentes especializados (dos capas)	Combinar similitud vectorial (Capa 1) y validación por LLM con prompt especializado (Capa 2), articulados en agentes con roles delimitados.	Aprovecha la infraestructura existente. La doble capa eleva la precisión. Los agentes separan responsabilidades y facilitan el mantenimiento.	Mayor complejidad que enfoques de capa única. Requiere definir umbrales de confianza para ambas capas.	**Seleccionado**

La primera alternativa explorada fue un sistema de RAG con preguntas diagnósticas personalizadas por cliente. Si bien esta opción ofrecía alta personalización, se identificó que un flujo extenso de preguntas afectaría negativamente la experiencia del cliente en un canal de mensajería inmediata y no sería eficiente como mecanismo de clasificación principal dado el volumen de interacciones. Sin embargo, el enfoque no fue descartado completamente: se reconoció su valor como mecanismo de apoyo en casos límite, y quedó incorporado parcialmente en la lógica del Agente Validador cuando la confianza de clasificación es insuficiente.

La segunda alternativa considerada fue la clasificación basada únicamente en similitud vectorial mediante *embeddings* del *dataset* histórico de ADMS. Aunque representa un enfoque rápido y directamente sustentado en el historial operativo de Celsia, se descartó como solución única porque la similitud vectorial no captura matices semánticos complejos ni tipologías poco representadas en el *dataset*, y porque desaprovechar la capacidad LLM ya instalada desde 2025 representaría un retroceso técnico injustificado.

A partir de estas evaluaciones, se seleccionó una tercera alternativa que combina ambos enfoques en dos capas de clasificación integradas en un sistema de agentes especializados. La hipótesis central planteada fue que, potenciando el flujo conversacional actual con agentes de mayor precisión semántica implementados sobre `LangGraph` y `CrewAI`, es posible reducir significativamente la proporción de conversaciones no clasificables y evitar que reportes ajenos a fallas reales de la red eléctrica continúen ingresando al flujo operativo de energía.

Sobre este principio se idearon cuatro agentes especializados: un Agente Clasificador que combina similitud vectorial sobre *embeddings* del *dataset* histórico de ADMS con el razonamiento de un LLM que, mediante un *prompt* especializado, valida o corrige la categoría sugerida, avanzando en el flujo solo cuando ambas capas convergen con un nivel de confianza suficiente; un Agente Router que determina si el reporte corresponde al flujo de energía o debe dirigirse a otro canal, cubriendo exhaustivamente las tipologías que el sistema de 2025 no discrimina; un Agente Validador que verifica la coherencia entre la descripción del cliente y la categoría asignada, formulando preguntas guiadas cuando la confianza es insuficiente; y un Agente Creador de Incidente que, una vez validada la clasificación, genera el registro en ADMS con los datos del cliente correctamente normalizados, resolviendo la brecha de integración identificada en la fase de empatía.

3.2.5 Fase 4 - Prototipar

La fase de prototipado parte del reconocimiento de que el flujo conversacional de Celsia ha atravesado dos iteraciones previas: el flujo original de 2022, basado en selección de menú sin validaciones semánticas, y el flujo mejorado de 2025, que incorporó LLMs y un primer sistema de clasificación por intenciones. El prototipo propuesto en este proyecto constituye una tercera iteración, diseñada específicamente para atacar la brecha residual que las versiones anteriores no han podido resolver. Sin embargo, llegar al diseño definitivo implicó a su vez un proceso interno de tres bocetos sucesivos, cada uno ajustado a partir de nueva información recogida con los especialistas de Celsia.

El primer boceto surgió a mediados de 2025, a partir de la exploración directa del portal web de reporte de daños y del canal LuzIA WhatsApp, combinada con una sesión de trabajo con el especialista de Operación del Centro de Control. En esa sesión se validaron las brechas operativas que él mismo gestionaba en su rol de supervisión: reportes mal clasificados que llegaban a ADMS, visitas fallidas recurrentes y la dificultad de discriminar desde el sistema si un reporte correspondía a una falla real de la red o a una solicitud de otro tipo. Con base en ese levantamiento y en las pruebas realizadas directamente sobre los canales digitales, se diagramó un primer boceto del flujo propuesto.

No obstante, al iniciar las sesiones con el especialista de experiencia al Cliente, el equipo identificó que hacia finales de 2025 ya se habían implementado mejoras significativas en el flujo conversacional de LuzIA que el primer boceto no contemplaba. El flujo en el que se había basado el diseño inicial había quedado desactualizado. A partir de la información provista por experiencia al cliente sobre la arquitectura real del sistema actualizado, se elaboró un segundo boceto que incorporó la infraestructura LLM vigente como punto de partida, redefiniendo el rol de los agentes propuestos como una capa adicional sobre lo existente y no como un reemplazo.

Finalmente, en las sesiones con el especialista de Tecnología, se revisaron los detalles técnicos de integración entre los agentes propuestos y el sistema ADMS, lo que permitió precisar tres aspectos clave del diseño: el límite de tres rondas de preguntas guiadas antes del *fall-back* al operador humano, la ubicación del Agente *Router* antes de cualquier escritura en ADMS, y la normalización de los campos de nombre, apellido y celular del cliente en el Agente Creador. Estos ajustes dieron lugar al boceto definitivo que se presenta en la sección 3.4.

La Tabla 8. Iteraciones del proceso de prototipado - flujo conversacional propuesto resume las tres iteraciones del proceso de prototipado, los insumos que motivaron cada ajuste y los cambios incorporados en cada versión.

Tabla 8. Iteraciones del proceso de prototipado - flujo conversacional propuesto

Iteración	Momento y fuente	Insumo recibido	Ajustes incorporados	Resultado
Boceto 1 Mediados 2025	**Exploración inicial de canales digitales y sesión con Darío Díaz Gómez (Operación)**	Pruebas directas sobre el portal web y LuzIA WhatsApp. Identificación de brechas operativas desde la perspectiva del Centro de Control: reportes mal clasificados, visitas fallidas recurrentes y dificultad para	Primer diagrama del flujo propuesto basado en el estado del sistema a mediados de 2025. Definición inicial de los roles de los agentes y los puntos de decisión del flujo.	Primer boceto del flujo *TO-BE* (basado en sistema pre-actualización 2025)

		discriminar tipologías desde el sistema.		
Boceto 2 Finales 2025	**Sesiones con Luis Carvajal (Experiencia al Cliente)**	Información sobre las mejoras implementadas en LuzIA hacia finales de 2025 que el primer boceto no contemplaba. El flujo inicial había quedado desactualizado respecto al sistema en producción.	Redefinición del punto de partida: los agentes propuestos pasan a operar como capa adicional sobre la infraestructura *LLM* existente, no como reemplazo. Actualización del flujo para reflejar la arquitectura real vigente.	Segundo boceto del flujo *TO-BE* (alineado con infraestructura *LLM* actualizada)
Boceto 3 (definitivo) 2026	**Sesiones con Sebastián (Tecnología)**	Revisión técnica de la integración entre los agentes propuestos y los sistemas internos. Validación de restricciones de diseño y puntos críticos de la arquitectura.	Incorporación del límite de tres rondas de preguntas antes del *fall-back*. Reubicación del Agente *Router* antes de cualquier escritura en el sistema de gestión.	Boceto definitivo presentado en la sección 3.4 y desarrollado en el Capítulo 4

Esta tercera iteración se materializa en el diagrama de flujo propuesto (TO-BE) presentado en la sección 3.4 y en el diseño conceptual de la arquitectura de agentes que se desarrolla en detalle en el Capítulo 4.

3.2.6 Fase 5 - Evaluar

La evaluación en esta fase se desarrolló en dos niveles: una validación conceptual del diseño propuesto, realizada de forma integrada a lo largo de las sesiones de levantamiento de información con los especialistas de Celsia, y una validación experimental del prototipo funcional, documentada en el Capítulo 5.

La validación conceptual se llevó a cabo mediante la contrastación progresiva del diseño con cada uno de los tres perfiles participantes, según el área de conocimiento de cada especialista. Con el especialista de Operación se validó que el principal impacto operativo de una clasificación incorrecta es el despacho de brigadas a sitios sin falla real, y que un mecanismo de filtro previo al despacho, como el que propone el Agente Router, respondería directamente al problema

gestionado desde el Centro de Control. Con el especialista de Experiencia al Cliente se validó que el enfoque de agentes especializados sobre la infraestructura existente era coherente con el principio operativo de Celsia de construir sobre lo que ya funciona. Con el especialista de Tecnología se validaron los puntos de integración entre los agentes propuestos y los sistemas internos, confirmando la viabilidad técnica del diseño antes de llevarlo a implementación.

Adicionalmente, el diseño del flujo TO-BE fue contrastado sistemáticamente contra tres referentes: los hallazgos consolidados de la fase de empatía, verificando que cada agente propuesto respondía a al menos una brecha identificada; el comportamiento conocido del sistema de 2025, para asegurar que la solución no replicaba limitaciones ya existentes; y las condiciones reales del canal digital, para garantizar que la arquitectura propuesta era compatible con el entorno operativo de LuzIA.

Como resultado de este proceso, se incorporaron al diseño final tres ajustes documentados en la fase de prototipado: el límite de tres rondas de preguntas guiadas antes del *fall-back* al operador humano, la ubicación del Agente Router antes de cualquier escritura en ADMS, y la normalización de los datos del cliente en el Agente Creador.

La validación experimental mediante métricas de precisión, tasa de aciertos y reducción efectiva del porcentaje no clasificable se documenta en el Capítulo 5, donde se presentan los resultados de las pruebas realizadas sobre el prototipo funcional en un entorno controlado.

3.3 Análisis de la evolución del flujo conversacional

El flujo de atención de reportes de daños LuzIA, ha atravesado dos iteraciones identificables desde su implementación en 2022, cada una respondiendo a los aprendizajes operativos de la etapa anterior.

3.3.1 Flujo inicial (2022) – Sin validación semántica

El flujo original, ilustrado en la Ilustración 2, presentaba al cliente un menú desplegable con múltiples opciones de motivo de solicitud, entre ellas sector sin servicio, poste caído, variación de voltaje, daño en transformador, línea reventada, solo mi predio sin energía, solicitud de reconexión, daño de internet, alumbrado público y medidor o acometida dañada. Algunas de estas opciones activaban directamente la creación del incidente en ADMS, mientras que otras desplegaban un modal de redirección indicando al cliente que su caso debía gestionarse por otro canal.

El problema de fondo no era la ausencia de opciones, sino la ausencia de validación entre lo que el cliente seleccionaba y lo que realmente describía en el campo de comentario libre. En la práctica, muchos clientes desconocían cuál opción correspondía a su situación y optaban por la más intuitiva, generalmente "Sector sin servicio", dejando en el comentario la descripción real de su problema: una solicitud de reconexión, un daño de internet, una queja por alumbrado público o una falla interna de su instalación. El sistema ignoraba por completo ese comentario y creaba el incidente en ADMS según la opción del menú, sin cruzar ambas fuentes de información. Esta

desconexión entre la selección del cliente y su descripción real era la causa raíz de la alta tasa de visitas técnicas fallidas documentada en el período 2022-2023.

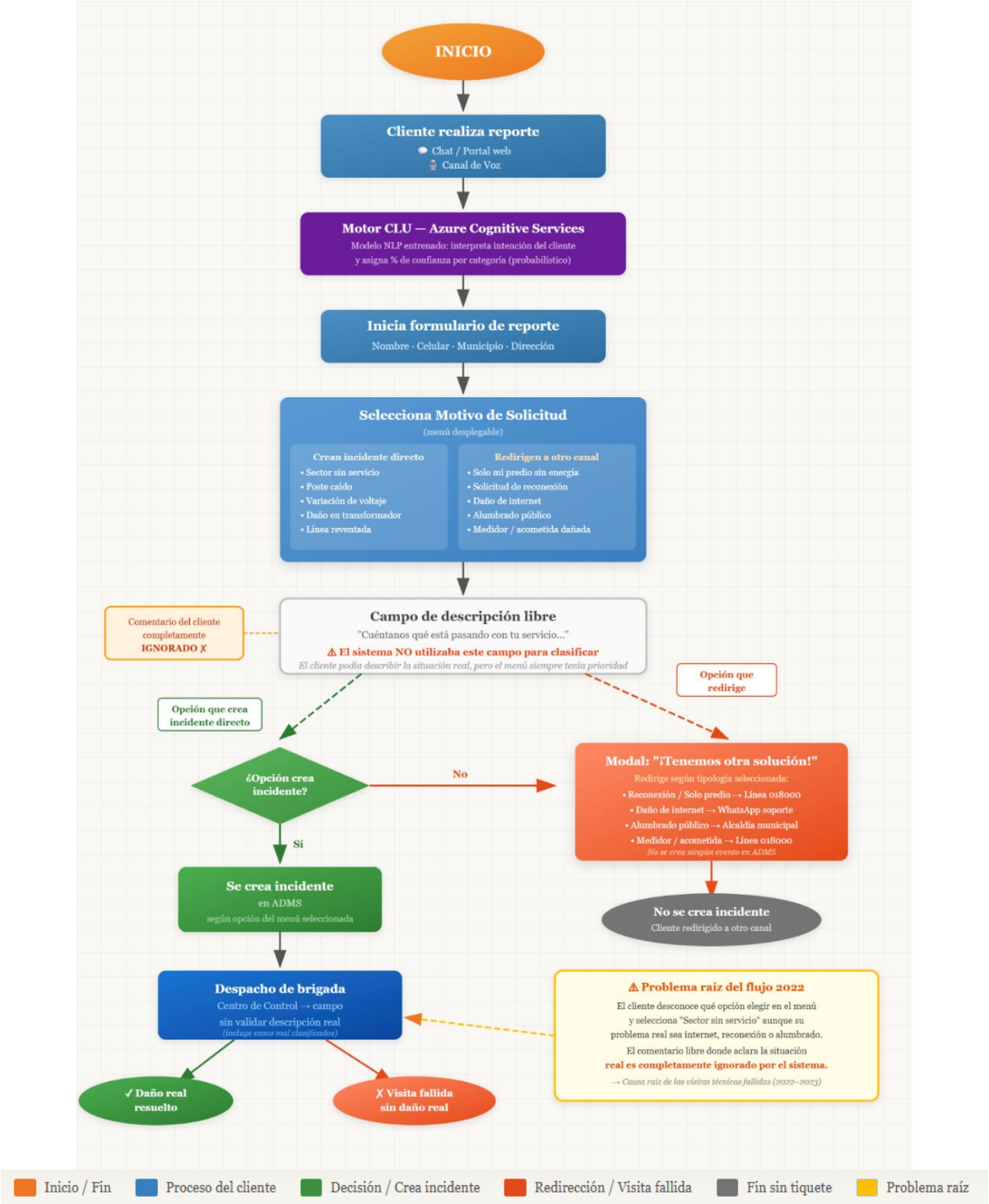


Ilustración 4. Flujo conversacional original (2022) - Sin validación semántica. Fuente: Elaboración propia a partir de sesiones de levantamiento con el equipo de Experiencia al Cliente de Celsia, diagramación asistida por Claude.

3.3.2 Flujo mejorado (2025) – Incorporación de LLM y agentes virtuales

A partir de agosto de 2025, Celsia rediseñó el ecosistema LuzIA migrando desde el motor CLU de Azure Cognitive *Services* hacia un sistema de clasificación basado en LLM conectados directamente a la API de Azure OpenAI, con Twilio como proveedor de integración para el canal de WhatsApp. Este cambio representó un salto cualitativo en la capacidad de comprensión del sistema: mientras el CLU asignaba porcentajes de confianza por categoría a partir de un modelo entrenado con reglas y ejemplos predefinidos, el nuevo LLM interpreta el texto libre del cliente de forma conversacional y devuelve directamente el *string* de clasificación correspondiente, operando con temperatura 0 y un máximo de 1.000 tokens para garantizar respuestas determinísticas y compatibles con los *templates* de WhatsApp.

El sistema opera con tres capas de inteligencia. Un primer LLM clasificador, de modelo básico y bajo costo, analiza el texto del cliente al inicio de la conversación y lo asigna a uno de seis flujos principales: reportar daño de energía, descargar factura, duplicado de factura, acuerdos de pago, puntos de venta y preguntas abiertas. Cuando este primer clasificador no logra asignar la intención con certeza, el caso pasa a un segundo LLM con capacidad de Recuperación Aumentada por Generación (RAG), que busca en la base de conocimiento del sistema para entregar una respuesta contextual más elaborada. Un tercer agente especializado en explicación de facturas atiende las consultas sobre cargos específicos en la cuenta del cliente.

Esta arquitectura elevó la efectividad de clasificación de aproximadamente el 40% que tenía el sistema CLU original hasta un 95%, reduciendo significativamente los incidentes mal clasificados en ADMS. Sin embargo, el sistema dejó abiertas dos brechas que motivaron el presente proyecto. La primera es que el 9,6% de las conversaciones permanece sin clasificar al corte de marzo de 2026: ante estos casos el sistema solo responde con un mensaje genérico de disculpa y remite al *call center*, sin extraer ningún dato operativo útil. La segunda es que algunas tipologías ajenas al servicio eléctrico, como aseo, alumbrado público, reclamaciones, reconexiones y clientes en zonas de peaje, continúan llegando al flujo de daños de energía cuando el cliente enmascara su solicitud real dentro de una descripción que el clasificador interpreta como un reporte de falla eléctrica. El flujo mejorado de 2025 se muestra en la Ilustración 5.

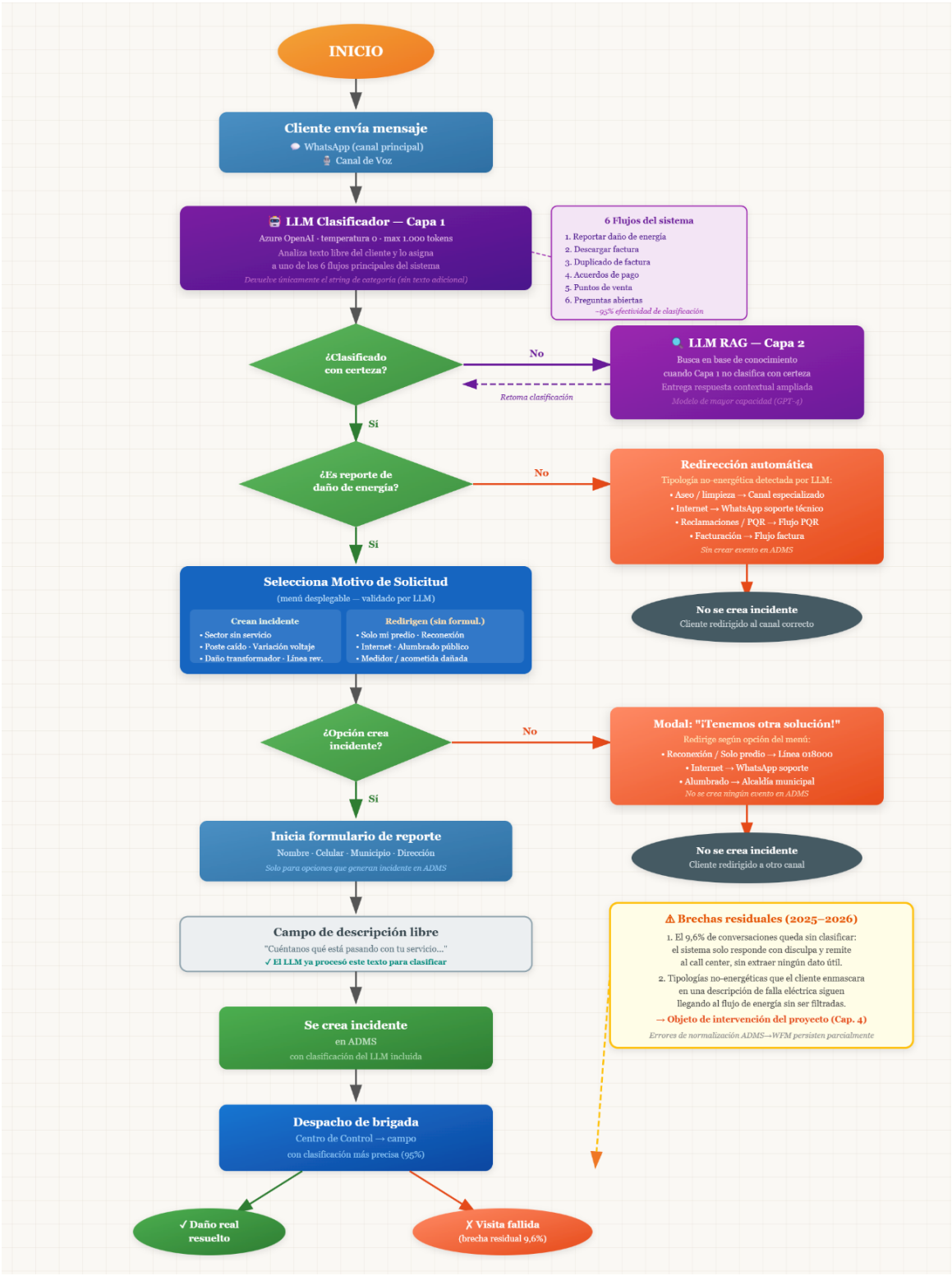


Ilustración 5. Flujo conversacional 2025 – Sistema LLM + RAG, Fuente: Elaboración propia, diagramación asistida por Claude.

3.4 Flujo conversacional propuesto (TO-BE)

El flujo propuesto no parte de cero ni reemplaza el sistema de 2025, toma como base la infraestructura LLM ya existente. Agrega un sistema de cuatro agentes especializados que resuelven específicamente las dos brechas que ese sistema no ha podido cerrar. La decisión de tomar como base la infraestructura existente, en lugar de replantearla toda, responde al principio operativo identificado en la fase de empatía, la arquitectura actual ya funciona para el 90,4% de las conversaciones; cambiarla representaría costos e incertidumbre innecesarios.

El flujo TO-BE se articula en cuatro etapas secuenciales ejecutadas por agentes especializados implementados sobre `LangGraph` y `CrewAI`:

Etapa 1 - Captura y análisis semántico (Agente Clasificador): cuando el cliente ingresa una descripción libre del problema, el Agente Clasificador procesa el texto mediante un enfoque híbrido de dos capas. En una primera capa el sistema genera *embeddings* del texto y realiza comparación por similitud vectorial contra el *dataset* histórico de reportes de Celsia. Posteriormente, una segunda capa, que incorpora un LLM que actúa como juez semántico. Este modelo valida o ajusta la clasificación inicial considerando el contexto completo del mensaje del cliente. Finalmente, el sistema combina ambas capas mediante un esquema de ponderación configurable, permitiendo balancear la influencia de los *embeddings* y del LLM en la decisión final. Este enfoque permite mejorar la precisión de la clasificación, evitando depender exclusivamente de reglas predefinidas o de la selección manual del usuario en un menú.

Etapa 2 - Enrutamiento inteligente (Agente Router): el Agente Router evalúa si el reporte corresponde a sector sin energía o a alguna de las tipologías que el flujo actual no discrimina correctamente, internet, alumbrado público, aseo, facturación, reclamaciones, reconexiones y peajes. Los reportes no energéticos se redirigen automáticamente al canal especializado correspondiente antes de generar ningún evento en ADMS, eliminando así la principal fuente de visitas fallidas.

Etapa 3 - Validación y confirmación (Agente Validador): para los reportes de sector sin energía, el Agente Validador verifica la coherencia entre la descripción del cliente y la categoría sugerida, aplicando un umbral de confianza. Cuando el nivel de confianza es insuficiente, el agente formula preguntas guiadas, como ¿el problema afecta solo su vivienda o también a sus vecinos? o ¿escucha zumbido en el transformador cercano?, para obtener información adicional. Si tras tres rondas de preguntas no se alcanza el umbral, se activa un mecanismo de *fall-back* que remite el caso al operador humano.

Etapa 4 - Creación del incidente (Agente Creador): una vez validada la clasificación, el Agente Creador genera el registro con los datos del cliente correctamente normalizados, nombre, apellido y número de celular en los campos correspondientes y con la clasificación validada incluida.

El diseño detallado de la arquitectura técnica de estos cuatro agentes, incluyendo los modelos

utilizados, los *prompts* de cada agente, la lógica del grafo de estados y los mecanismos de *fall-back*, se presenta en el Capítulo 514.

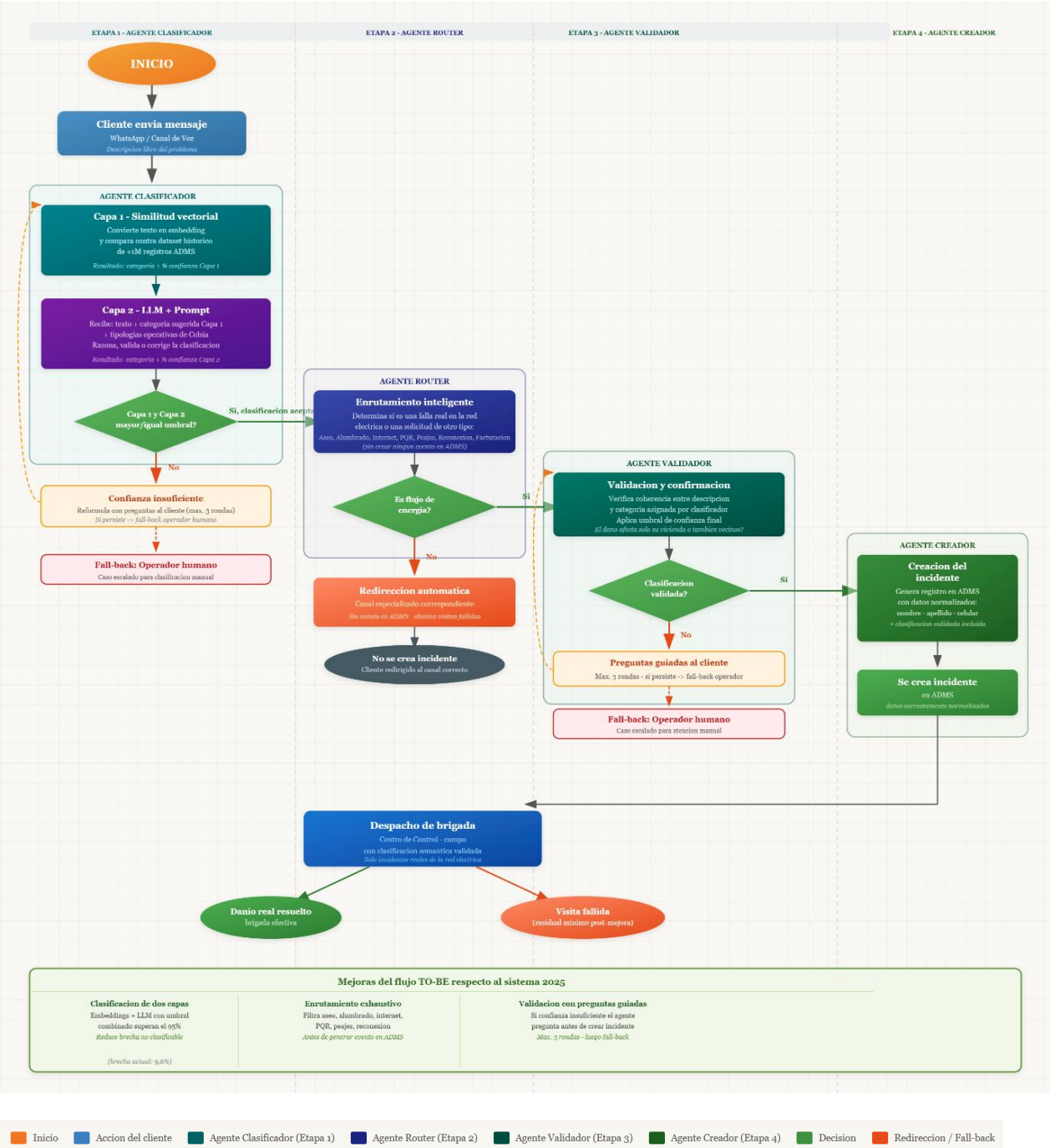


Ilustración 6. Flujo TO-BE, Fuente: Elaboración propia, diagramación asistida por Claude.

3.5 Comparación AS-IS / TO-BE

La **¡Error! No se encuentra el origen de la referencia.** 9 sintetiza las diferencias entre las tres versiones del flujo conversacional: el flujo original de 2022, el flujo mejorado de 2025 (AS-IS) y la propuesta de 2026 (TO-BE). Las celdas marcadas (a validar Cap.0) corresponden a resultados que se medirán en la fase de validación experimental.

Tabla 9. Comparación entre el flujo actual y el flujo propuesto

Dimensión	Flujo 2022	Flujo 2025 (AS-IS)	Propuesta 2026 (TO-BE)
Elemento primario	Tipo de falla (menú desplegable)	Descripción libre + menú	Descripción libre procesada por NLP (sin menú previo)
Motor de clasificación	CLU — *Azure Cognitive Services* (modelo NLP probabilístico)	LLM — Azure OpenAI (clasificador + RAG)	LLM existente + Agente Clasificador (*embeddings* + *prompt* especializado)
Validación semántica	Ninguna. El comentario libre era ignorado; el menú tenía prioridad absoluta.	Básica: coherencia tipo/descripción. Efectividad: ~95%	NLP con dos capas: similitud vectorial + LLM. Solo avanza si ambas capas superan el umbral de confianza.
IA generativa	No	Sí, preguntas de respuesta abierta (ago. 2025)	4 agentes especializados sobre `LangGraph` / `CrewAI`
Enrutamiento	Único flujo para todos los reportes, sin distinción de tipología	Parcial (pendiente detalle TI)	Diferenciado por tipología completa antes de generar evento en ADMS
Tipologías distintas a energía cubiertas	Ninguna. Solo modal de redirección si el cliente seleccionaba la opción correcta.	Parcial (pendiente detalle TI)	Internet, alumbrado, aseo, facturación, reclamaciones, peajes, reconexión
% no	Alto (sin dato	9,6% (mar. 2026)	Reducción estimada (a

Dimensión	Flujo 2022	Flujo 2025 (AS-IS)	Propuesta 2026 (TO-BE)
clasificable	histórico)	Responde con disculpa y remite al *call center*	validar Cap. 5)
Priorización de incidentes	Estándar para todos	Estándar para todos	P1/P2/P3 automático por IA según tipo y severidad de la falla
Intervención humana	Equipo dedicado de clasificación manual (alto costo operativo)	Reducida	Solo casos escalados (*fall-back* tras 3 rondas sin certeza)
Visitas fallidas	Sin mecanismo de prevención. Causa raíz: comentario ignorado	Reducidas parcialmente	Filtradas antes del despacho mediante validación semántica en dos capas
Costo estimado de operación	Alto. Equipo humano dedicado a clasificación manual más costos por visitas fallidas no filtradas. Costo estimado por visita fallida: $90.000 COP.	Medio. Se eliminó el equipo de clasificación manual, pero persisten costos por el 9,6% de conversaciones no resueltas y por visitas fallidas no prevenidas.	Reducción estimada del 20% al 40% en costos asociados a visitas fallidas. Los agentes operan sobre infraestructura LLM existente sin requerir equipo humano de clasificación (a validar Cap. 5).
Tiempo de implementación	4 meses de piloto controlado previo al escalamiento (sep. 2022 – ene. 2023).	Implementación incremental sobre la arquitectura CLU existente. Migración al LLM completada hacia mediados de 2025.	Construcción sobre infraestructura LLM ya instalada. No requiere reemplazo de componentes existentes, lo que reduce el tiempo de integración respecto a una arquitectura desde cero (estimación a detallar en Cap. 4).

Fuente: Elaboración propia a partir de datos operativos de Celsia y sesiones de levantamiento con los equipos de Experiencia al Cliente y Tecnología (2026).

3.6 Beneficios esperados del flujo propuesto

Respecto al flujo de 2025, la propuesta busca producir los siguientes resultados operativos incrementales:

- Reducción del porcentaje de conversaciones no clasificables, actualmente en 9,6%, mediante agentes con mayor capacidad semántica y preguntas guiadas ante baja confianza en la clasificación.
- Cobertura exhaustiva de reportes ajenos a fallas reales de la red eléctrica que el flujo actual no redirige correctamente, eliminando su ingreso al flujo operativo de energía antes de que generen un evento en ADMS.
- Reducción adicional de visitas técnicas fallidas al fortalecer el filtro semántico antes del despacho de brigadas, con el consecuente impacto positivo en los indicadores SAIDI y SAIFI.
- Mejora de la experiencia del cliente mediante respuestas más precisas, tiempos estimados de atención basados en localización y tipo de falla validado, y reducción del número de interacciones necesarias para completar un reporte.

Los resultados cuantitativos de estos beneficios esperados se medirán y documentarán en el Capítulo 0, mediante la validación experimental del prototipo

4. CONSTRUCCIÓN DE SISTEMA INTELIGENTE DE CLASIFICACIÓN DE REPORTES

Este capítulo presenta el diseño de la arquitectura del sistema inteligente de clasificación de reportes de incidentes, los roles de cada agente que lo compone, el mecanismo de *fall-back* ante clasificaciones de baja confianza, y los avances en la preparación del *dataset* histórico que constituye su insumo fundamental.

El diseño no parte de cero: toma como base el sistema de agentes con IA generativa implementado por Celsia en 2025 y añade una capa de especialización semántica orientada a resolver la brecha residual del 9,6% de conversaciones no clasificables identificada en el capítulo anterior. El prototipo fue construido e implementado en Python, conectado a un bot de Telegram público (@Luzia_celsia_bot) y a una interfaz web.

El objetivo de este capítulo es documentar las decisiones de diseño, los criterios técnicos adoptados y los resultados obtenidos en el procesamiento de datos, que constituyen la base para la implementación del sistema de clasificación automática.

4.1 Arquitectura general del sistema

El sistema propuesto se concibe como una arquitectura de cinco capas funcionales que operan de manera orquestada: el canal digital de entrada, el orquestador central basado en `LangGraph`, los agentes especializados implementados con `CrewAI`, el mecanismo de *fall-back* y los sistemas operativos de destino.

El prototipo implementado opera sobre dos canales simultáneos: un bot público de Telegram (@Luzia_celsia_bot), accesible a cualquier usuario con la aplicación instalada, y una interfaz web desarrollada con FastAPI que replica el flujo conversacional del sistema real de Celsia, incluyendo el menú de motivos de reporte como primer filtro de enrutamiento. [Ver Ilustración 7]

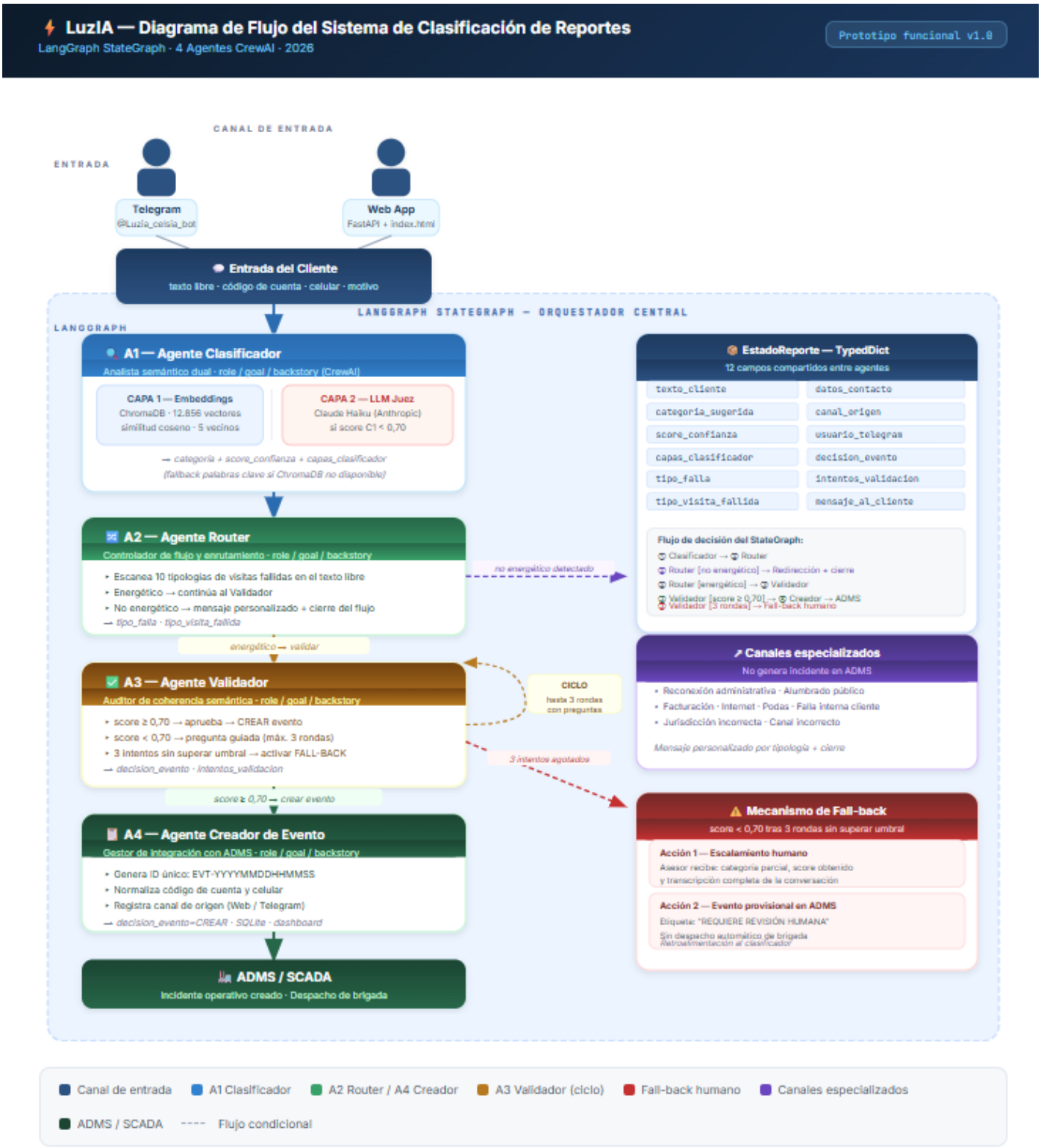


Ilustración 7. Arquitectura conceptual del sistema inteligente de clasificación de reportes de daños. Fuente: Elaboración propia diagramación asistida por Claude.

Capa 1 - Canal digital de entrada

El sistema recibe reportes a través de dos canales: el bot de Telegram @Luzia_celsia_bot (accesible públicamente a cualquier usuario) y la interfaz web, desarrollada con FastAPI y diseñada con el mismo flujo conversacional que el sistema actual de Celsia. En ambos canales el cliente proporciona: motivo del reporte (seleccionado de un menú con las categorías del *dataset*), código de cuenta (identificador del cliente en ADMS), número de celular y descripción libre del problema.

Capa 2 - Orquestador central: LangGraph StateGraph

- El orquestador es el componente central de la arquitectura. Se implementa como un `StateGraph` de `LangGraph`: un grafo de estado dirigido que coordina el flujo entre los agentes mantiene el estado persistente de la conversación y controla las rutas de ejecución mediante aristas condicionales. El estado del grafo se define como un `TypedDict` de Python con doce campos que acompañan al reporte a lo largo de toda la ejecución. Ver Tabla 10. Campos del Estado Reporte - TypedDict compartido por los cuatro agentes en el StateGraph. Fuente: Elaboración propia..

Tabla 10. Campos del Estado Reporte - TypedDict compartido por los cuatro agentes en el StateGraph. Fuente: Elaboración propia.

CAMPO	TIPO	DESCRIPCIÓN
texto_cliente	str	Descripción libre ingresada por el cliente
categoria_sugerida	str	Salida del Agente Clasificador (ej. SECTOR_SIN_ENERGIA)
score_confianza	float	Valor 0-1 que determina el flujo en el Validador
capas_clasificador	dict	Detalle de capa 1 y capa 2: categoría, score, convergencia, modo
tipo_falla	str	"energetico" o "no_energetico" - salida del Router
tipo_visita_fallida	str	Tipología detectada por el Router (PODAS, FACTURACION, etc.)
datos_contacto	dict	codigo_cuenta, celular del cliente
canal_origen	str	"Web" o "Telegram" — identifica el canal de entrada
usuario_telegram	str	@username o Usuario_{ID} cuando el canal es Telegram
decision_evento	str	"CREAR", "FALLBACK_HUMANO" o vacío si no energético
intentos_validacion	int	Contador de rondas de preguntas guiadas (máx. 3)
mensaje_al_cliente	str	Texto que el sistema devuelve al cliente en cada paso

La capacidad de `LangGraph` de modelar ciclos en el grafo es fundamental para implementar el comportamiento iterativo del agente validador, que puede solicitar información adicional al cliente antes de tomar una decisión de clasificación definitiva [37][38].

Capa 3 - Agentes especializados: `CrewAI`

Los cuatro agentes del sistema se implementan con `CrewAI`, cada uno con un rol, objetivo y trasfondo definido explícitamente. Su descripción detallada se presenta en la sección 4.2.

Capa 4 - Mecanismo de *fall-back*

Gestiona los casos en que el sistema no logra clasificar con suficiente confianza. Su descripción detallada se presenta en la sección 4.3.

Capa 5 - Sistemas operativos de destino

Los reportes de energía validados se envían al sistema ADMS/SCADA para la creación del incidente operativo. Los reportes que no corresponden a fallas reales en el servicio de energía eléctrica se redirigen a los canales especializados correspondientes (reconexión, internet, alumbrado público, facturación, etc.) sin generar incidente en ADMS, eliminando así el origen de las visitas técnicas fallidas documentadas en el EDA sección 4.4.2.

4.2 Roles y responsabilidades de cada agente

El sistema está compuesto por cuatro agentes especializados. La secuencia de actuación es invariable: clasificar → enrutar → validar → registrar. Cada agente recibe el estado compartido, lo actualiza con su resultado y lo transfiere al siguiente nodo del grafo.

Tabla 11. Flujo de decisión del StateGraph: nodos, lógica y salidas de cada agente. Fuente: Elaboración propia.

NODO	LÓGICA DE DECISIÓN Y SALIDA
ENTRADA	Cliente escribe motivo + descripción libre + código de cuenta + celular
A1 CLASIFICADOR	Capa 1: *embeddings* + ChromaDB (12.856 vectores) Capa 2: LLM juez (Claude Haiku / GPT-4o-mini evaluados) si score < 0.70 (definido por diseño · validación empírica en Cap. 5) → Devuelve: categoría + score + capas_clasificador
A2 ROUTER	¿Tipología de visita fallida detectada en texto? → no_energetico + mensaje personalizado ¿Categoría energética? → pasa a Validador ¿NO_ENERGETICO sin tipología? → DATO_INSUFICIENTE
A3 VALIDADOR	score ≥ 0.70 → aprueba (CREAR) score < 0.70, intentos < 3 → pregunta guiada (REINTENTAR) intento 3 sin superar umbral → FALLBACK_HUMANO
A4 CREADOR	Genera evento con ID: EVT-YYYYMMDDHHMMSS Normaliza código de cuenta y celular Registra canal de origen (Web / Telegram) Guarda en SQLite + muestra en dashboard

FALLBACK	Transfiere con contexto completo al asesor humano Registra en SQLite para retroalimentación Crea evento provisional con etiqueta REQUIERE REVISIÓN

Agente 1 - Clasificador (A1)

Rol: Analista semántico de reportes de incidentes eléctricos.

Objetivo: Determinar la naturaleza real del incidente mediante un mecanismo de clasificación de dos capas que combina similitud vectorial sobre el *dataset* histórico de Celsia con validación por LLM.

Mecanismo: El Agente Clasificador opera en dos capas sucesivas cuya activación depende del score de confianza obtenido:

Tabla 12. Mecanismo dual del Agente Clasificador: capas de embeddings, LLM juez y fallback de palabras clave. Fuente: Elaboración propia.

CAPA	MECANISMO	CONDICIÓN DE AVANCE
CAPA 1 Embeddings + ChromaDB	Modelo: paraphrase-multilingual-MiniLM-L12-v2 *Dataset*: 12.856 vectores (EDA histórico de Celsia) Métrica: similitud coseno · 5 vecinos más cercanos Voto mayoritario → categoría ganadora + score	score ≥ 0.70 → Resultado final (no activa C2)
CAPA 2 LLM Juez (Claude Haiku / GPT-4o-mini)	Se activa solo si score C1 < 0.70 Modelos evaluados: claude-haiku-4-5-20251001 (Anthropic) · gpt-4o-mini (Azure OpenAI) *Prompt* especializado con categorías Celsia Convergencia: misma cat. → score promedio Divergencia: LLM gana, score = 0.65	score C2 ≥ 0.70 → Resultado final Divergencia → score 0.65
FALLBACK Palabras Clave	Se activa si ChromaDB no está disponible Diccionario de *keywords* por categoría Score máximo = 0.74 (fuerza activación de C2) Garantiza operación sin interrupción	Siempre activa C2 como juez (score cap 0.74)

La capa de *fallback* por palabras clave garantiza que el sistema opere sin interrupción incluso si ChromaDB no está disponible, limitando el score a 0,74 para forzar la activación del LLM juez como segunda capa de validación. Esta decisión de diseño asegura que ningún reporte se clasifique únicamente con el mecanismo menos preciso.

Insumo clave: *dataset* histórico de 88.944 registros depurados, vectorizado en ChromaDB con 12.856 embeddings distribuidos estratégicamente para compensar el desbalance de clases (ver sección 4.4).

Agente 2 - Router (A2)

Rol: Controlador de flujo y enrutamiento multicanal.

Objetivo: Determinar si el reporte corresponde a una falla real en la red eléctrica de Celsia o si debe redirigirse a un canal especializado antes de generar un incidente en el sistema ADMS.

Mecanismo: El Router opera en dos fases: primero escanea el texto libre en busca de palabras clave asociadas a tipologías de visitas fallidas resultantes del EDA; si detecta una tipología, redirige inmediatamente sin consultar la categoría del Clasificador. Este diseño corrige un sesgo identificado en las pruebas: el Clasificador puede etiquetar como RIESGO_ANOMALIA un reporte sobre podas (por la presencia de palabras como 'cables' y 'contacto'), pero el Router lo intercepta correctamente por la presencia de 'árbol' o 'poda'.

Las tipologías implementadas provienen directamente del análisis de los 5.111 registros de visitas fallidas resultantes del EDA:

Tabla 13. Tipologías de visitas fallidas detectadas por el Router, con base en el EDA de 5.111 registros históricos de Celsia. Fuente: Elaboración propia.

TIPOLOGÍA	CASOS EDA	PALABRAS CLAVE / DETECCIÓN	CANAL DE REDIRECCIÓN
PODAS	1.201	*poda, árbol, rama, vegetación, contacto con líneas*	Área mantenimiento de redes
DATO_INSUFICIENTE	962	*Texto sin palabras clave reconocibles*	Canal especializado
FALLA_INTERNA_CLIENTE	900	*solo mi casa, solo mi apartamento, breaker, taco*	Electricista certificado
SERVICIO_AJENO	565	*internet, televisión, gas, acueducto, teléfono*	Operador del servicio
SIN_HALLAZGO_TRANSITORIA	464	*detectada por semántica contextual*	Canal especializado
ALUMBRADO_PUBLICO	339	*alumbrado, lámpara, luminaria, calle oscura*	Alcaldía / municipio
CANAL_INCORRECTO	295	*reclamo, queja, PQRS, solicitud formal*	Línea de atención / web
RECONEXION_ADMINISTRATIVA	209	*reconexión, reconectar, me cortaron, corte adm.*	Línea de cartera
FACTURACION	151	*factura, cobro, pago, deuda, recibo*	Línea de atención / web
JURISDICCION_INCORRECTA	21	*EPM, EMCALI, Codensa, otra empresa*	Operador correspondiente

Impacto esperado: eliminación del ingreso al flujo de ADMS de los reportes que históricamente generaban visitas técnicas fallidas, identificados como la principal causa

de la brecha residual del 9,6%.

Agente 3 - Validador (A3)

Rol: Auditor de coherencia semántica y gestor de confirmación iterativa.

Objetivo: Garantizar que solo ingresen al sistema ADMS reportes con clasificación confiable, mediante un ciclo iterativo de hasta tres rondas de preguntas guiadas por categoría.

Mecanismo: El umbral de confianza se establece en 0,70, valor definido a partir del comportamiento esperado del mecanismo de clasificación dual: cuando las capas C1 y C2 divergen en la categoría asignada, el score final queda fijado en 0,65 por diseño, dado que en ese escenario el LLM juez prevalece, pero la convergencia no se alcanza. Un umbral de 0,75 rechazaría sistemáticamente estos casos, aunque la categoría resultante sea correcta. El umbral de 0,70 permite que reportes con divergencia entre capas, pero con categoría válida, avancen al Agente Creador de Evento sin activar innecesariamente el *fall-back*.

Las preguntas guiadas están definidas por categoría y se formulan de forma progresiva en cada ronda:

- SECTOR_SIN_ENERGIA: ¿Tus vecinos también están sin luz o es solo tu casa? / ¿Cuánto tiempo llevas sin servicio? / ¿Revisaste el interruptor principal?
- PROBLEMAS_VOLTAJE: ¿Con qué frecuencia ocurre la fluctuación? / ¿Qué electrodomésticos se ven afectados? / ¿El problema afecta solo tu casa?
- RIESGO_ANOMALIA: ¿Hay cables caídos, humo o chispas? / ¿Hay riesgo inmediato para personas? / ¿El incidente sigue ocurriendo?
- SUMINISTRO_PARCIAL: ¿Qué circuitos tienen luz y cuáles no? / ¿Cuánto tiempo lleva el suministro intermitente? / ¿Afecta a todo el sector?

Agente 4 -Creador de Incidente (A4)

Rol: Gestor de integración con el sistema ADMS.

Objetivo: Generar el registro del incidente con datos completos y normalizados, garantizando la integridad de la información que ingresa al sistema operativo.

Mecanismo: El agente construye el *payload* del evento con los siguientes campos: identificador único (EVT-YYYYMMDDHHMMSS), fecha y hora de creación, categoría validada, código de cuenta del cliente, celular en formato estándar, descripción original del cliente, estado del evento (ABIERTO) y canal de origen (Web o Telegram). El registro queda almacenado en la base de datos SQLite local y visible en tiempo real en el *dashboard* de monitoreo.

4.3 Mecanismo de *fall-back*

El mecanismo de *fall-back* garantiza que ningún reporte se pierda cuando el sistema no alcanza el umbral de confianza requerido. Su diseño responde al principio operativo de Celsia de no generar visitas fallidas, pero tampoco desestimar reportes que podrían corresponder a incidentes reales.

Condición de activación

El fall-back se activa cuando el Agente Validador no alcanza un score ≥ 0,70 tras tres rondas de preguntas guiadas, o cuando el cliente no proporciona información suficiente para mejorar el score de confianza.

Acciones del mecanismo

- **Acción 1- Escalamiento a asesor humano:** el reporte se transfiere automáticamente a un asesor del centro de control, acompañado del contexto completo de la conversación, la categoría parcialmente inferida, el score obtenido y el canal de origen. El asesor toma la decisión final con información de mejor calidad que la disponible en el flujo manual actual.
- **Acción 2 -Evento provisional:** si no hay asesor disponible, se crea un evento en ADMS con la observación encabezada en mayúsculas por la etiqueta "REQUIERE REVISIÓN HUMANA", seguida del contexto parcial de la conversación y el score de confianza obtenido. Esta convención garantiza que el evento sea identificable en la cola de revisión operativa sin generar despacho automático de brigada.
- **Retroalimentación para reentrenamiento:** todos los casos de *fall-back* quedan registrados en SQLite con su resolución final. Este registro alimenta un proceso de reentrenamiento periódico del modelo clasificador, mejorando progresivamente la tasa de clasificación automática exitosa.

4.4 Procesamiento y preparación del *dataset*

Como paso previo indispensable para la construcción del sistema de clasificación, se trabajó con el *dataset* histórico de reportes provenientes de los canales digitales de atención de Celsia. Este proceso permitió comprender la naturaleza y distribución de los datos disponibles, identificar los patrones lingüísticos asociados a visitas técnicas fallidas y preparar las bases de información que alimentan dos de los cuatro agentes del sistema: el Clasificador y el Router.

La calidad del *dataset* determina directamente la precisión del sistema: un clasificador entrenado sobre datos ruidosos o mal etiquetados producirá scores de confianza sistemáticamente bajos, incrementando la tasa de *fall-back.* Por ello, el procesamiento no es un paso opcional sino una condición necesaria para el funcionamiento del sistema.

4.4.1 Limpieza y normalización

El proceso de preparación inició con el dataset bruto extraído del sistema ADMS, compuesto por registros de eventos reportados a través de los canales digitales de Celsia. El pipeline de limpieza se ejecutó en etapas secuenciales, tal como se muestra en la Tabla 14:

Tabla 14. Pipeline de limpieza y depuración del dataset histórico. Fuente: Elaboración propia

Etapa del pipeline	Registros	Descripción
Dataset original (ADMS)	89.943	*registros brutos extraídos*
Registros de prueba eliminados	999	*simulaciones y cancelaciones administrativas*
Registros con texto nulo	98	*campo texto_normalizado vacío o nulo*
Dataset final limpio	88.944	*registros disponibles para vectorización*
Visitas fallidas detectadas	5.111	*5,7% del dataset , insumo del Agente Router*

Las actividades de preparación del *dataset* incluyeron:

- Limpieza y normalización del texto libre: eliminación de caracteres especiales, normalización de tildes y estandarización de mayúsculas y minúsculas mediante scripts en Python con la librería pandas.
- Exclusión de 999 registros de prueba del sistema, simulaciones internas y cancelaciones administrativas, identificados por patrones en el campo CAUSE del sistema ADMS.
- Eliminación de 98 registros con campo texto_normalizado vacío o nulo, que no aportarían información semántica al modelo.
- Detección y anonimización de datos sensibles (nombres propios, números telefónicos y direcciones) en cumplimiento de la Ley 1581 de 2012 y su Decreto Reglamentario 1377 de 2013 [52]. El proceso generó un log de 88.709 sustituciones documentadas en el archivo log_anonimizacion.csv, reemplazando información personal por etiquetas genéricas ([NOMBRE], [TELEFONO], [DIRECCION]).
- Análisis de frecuencia de palabras y frases recurrentes, identificando patrones genéricos y expresiones ambiguas que dificultan la clasificación automática.

Este procesamiento evidenció una inconsistencia estructural en los datos, los campos más frecuentes del texto libre no son descriptores semánticos del problema eléctrico sino campos de captura de datos del cliente (dirección, teléfono, nombre), que concentran las primeras posiciones del ranking de frecuencia. La tabla siguiente ilustra este hallazgo:

Tabla 15. Términos más frecuentes en el dataset histórico y su relevancia semántica. Fuente: Elaboración propia.

Término	Frecuencia	Observación
dirección	258.614	Campo estructurado - baja discriminación semántica
teléfono	246.092	Campo estructurado - baja discriminación semántica
nombre	152.314	Campo estructurado - baja discriminación semántica
falla	46.055	Alta relevancia semántica — presente en todas las categorías
sector	26.763	Indica alcance geográfico del reporte
daño	18.358	Indica naturaleza del problema
voltaje	(en contexto)	Discriminador de PROBLEMAS_VOLTAJE
poda / árbol	(en contexto)	Discriminador de visita fallida tipo PODAS

Este hallazgo justifica la decisión de usar *embeddings* semánticos en lugar de modelos basados en frecuencia de palabras (como TF-IDF o *Bag of Words*): los términos más frecuentes son ruido estructural, no señal discriminativa. Un modelo de frecuencia clasificaría prácticamente todo como la categoría dominante. Los *embeddings* capturan el significado contextual del texto más allá de las palabras individuales, lo que los hace robustos ante este tipo de ruido.

El procesamiento también evidenció inconsistencias recurrentes entre el tipo de falla seleccionado por el cliente en el menú del sistema y el detalle textual suministrado, lo que el presente proyecto denomina 'brecha semántica', así como combinaciones de texto históricamente asociadas a visitas técnicas fallidas.

4.4.2 Análisis exploratorio del *dataset* (EDA)

El análisis exploratorio se realizó sobre dos conjuntos de datos complementarios, con roles distintos dentro del sistema.

Dataset principal: dataset_final_limpio.csv

El *dataset* depurado contiene 88.944 registros distribuidos en cinco categorías según la columna REASON del sistema ADMS. El análisis reveló un desbalance extremo entre clases, con implicaciones directas sobre la estrategia de vectorización. Ver Tabla 16

Tabla 16. Distribución de clases en el dataset depurado (n = 88.944). La clase dominante concentra el 91,2% de los registros. Fuente: Elaboración propia.

Categoría REASON	Registros	%
Sector sin energía	81.077	**91.2%**
Problemas de voltaje	4.308	**4.8%**
Riesgo o Anomalía	3.241	**3.6%**
Sector con suministro parcial	285	**0.3%**
Servicio restaurado	33	**0.04%**

El desbalance identificado es de naturaleza extrema: la categoría 'Sector sin energía' supera en más de 18 veces a la segunda categoría más frecuente. Este nivel de desbalance tiene consecuencias directas sobre el clasificador: sin una estrategia de compensación, el modelo aprendería a predecir casi siempre la clase dominante, obteniendo una *accuracy* artificialmente alta pero una capacidad discriminativa muy baja para las categorías minoritarias. La estrategia de compensación adoptada se describe en la sección 4.4.3.

Análisis del desbalance entre clases

Para cuantificar el impacto del desbalance se calculó la razón de desbalance entre la clase mayoritaria y cada clase minoritaria:

- Sector sin energía vs. Problemas de voltaje: razón 18,8:1
- Sector sin energía vs. Riesgo o Anomalía: razón 25,0:1
- Sector sin energía vs. Sector con suministro parcial: razón 284,5:1
- Sector sin energía vs. Servicio restaurado: razón 2.456,9:1

Estos valores ubican el problema en la categoría de desbalance severo (razones mayores a 10:1), para el cual la literatura recomienda técnicas de remuestreo o ajuste de pesos. La estrategia seleccionada fue *undersampling* de la clase dominante, limitada a 5.000 muestras, equilibrando la representación en el espacio vectorial sin perder información de las clases minoritarias.

Ejemplos anonimizados por categoría

La siguiente tabla presenta textos representativos de cada categoría, extraídos directamente del *dataset* depurado y anonimizados. Estos ejemplos ilustran los patrones lingüísticos característicos de cada tipo de reporte y permiten comprender el nivel de ambigüedad semántica que enfrenta el clasificador.

Tabla 17. Ejemplos anonimizados de textos por categoría REASON. Fuente: Elaboración propia a partir del dataset histórico de Celsia.

Sector sin energía	
Ej. 1	*"nombre cel telefono sector vereda rosa finca alabamas lerida tolima"*
Ej. 2	*"cliente reporta sector sin servicio desde esta mañana. todo el barrio afectado"*
Ej. 3	*"no hay energía desde anoche. todos los vecinos reportan lo mismo"*
Problemas de voltaje	
Ej. 1	*"fluctuacion baja solo prende bombillo. daño electrodomesticos urgente revision"*

Ej. 2	*"altos voltajes vivienda 250 danaron tv nevera solicitan revision urgente sector"*
Ej. 3	*"casa pasan lineas voltage llego toco bajar tacos es urgente"*
Riesgo o Anomalía	
Ej. 1	*"solicitud reubicacion poste. cables cerca de las viviendas riesgo alto"*
Ej. 2	*"requiere poda arboles encuentran contacto lineas electricas peligro inminente"*
Ej. 3	*"transformador emitiendo ruido extrano humo visible zona residencial"*
Sector con suministro parcial	
Ej. 1	*"comunica tienen sector algunas casas solicitan colaboracion novedad presentada"*
Ej. 2	*"algunas viviendas tienen servicio otras no. problema intermitente"*

El análisis de los ejemplos evidencia que el lenguaje de los clientes es informal, fragmentado y frecuentemente ambiguo. Un mismo reporte puede describir una falla de infraestructura en términos que semánticamente se asemejan a una pérdida de servicio ('sector sin energía'). Esta ambigüedad justifica el uso del LLM juez como segunda capa del clasificador cuando el score de similitud vectorial no supera el umbral de confianza.

Longitud promedio de los textos libres

Se analizó la distribución de la extensión de los textos normalizados para definir los parámetros de tokenización del modelo de *embeddings*:

Tabla 18. Estadísticas descriptivas de longitud de textos normalizados (n = 88.846 registros con texto válido). Fuente: Elaboración propia.

Mínimo	Máximo	Media	Mediana
3 caracteres	971 caracteres	**150,3 caracteres**	154,0 caracteres

Tabla 19. Distribución de longitud de textos normalizados por rango de caracteres. El 79,4% de los textos se concentra entre 101 y 200 caracteres. Fuente: Elaboración propia.

Rango de longitud	Registros	%
0 – 50 caracteres	1,395	1.6%
51 – 100 caracteres	10,575	11.9%
101 – 200 caracteres	**70,551**	**79.4%**
201 – 300 caracteres	5,729	6.4%
> 300 caracteres	596	0.7%

La concentración del 79,4% de los textos en el rango de 101 a 200 caracteres es una característica

favorable para la vectorización: el modelo paraphrase-multilingual-MiniLM-L12-v2 tiene un límite de 512 tokens (~384 caracteres), por lo que el 99,3% de los textos del *dataset* se vectoriza sin truncamiento. Los 596 registros con más de 300 caracteres representan el 0,7% del total y son susceptibles de truncamiento parcial, lo que se considera aceptable para el propósito del sistema.

Dataset de visitas fallidas: visitas_fallidas_detectadas.csv

El segundo conjunto de datos, identificado mediante análisis con IA sobre el *dataset* principal, contiene 5.111 registros etiquetados como visitas técnicas fallidas. Estos son eventos que generaron el desplazamiento de una brigada operativa sin encontrar una falla real en la red eléctrica. El análisis de este *dataset* es el insumo directo para el diseño del Agente Router.

Un hallazgo crítico del análisis es que el 75,7% de las visitas fallidas estaba registrado en ADMS con la categoría 'Sector sin energía', es decir, el cliente reportó pérdida de servicio, pero la brigada no encontró falla en la red. Este dato confirma que la categoría REASON asignada en ADMS no es suficiente para predecir si un reporte generará una visita fallida, y justifica la necesidad de analizar el texto libre del cliente más allá de la categoría.

Tabla 20. Distribución de las 5.111 visitas fallidas por tipología detectada. Las tres primeras tipologías concentran el 59,9% de los casos. Fuente: Elaboración propia.

Tipología de visita fallida	Casos	%
PODAS	1,201	**23.5%**
DATO_INSUFICIENTE	962	**18.8%**
FALLA_INTERNA_CLIENTE	900	**17.6%**
SERVICIO_AJENO	565	11.1%
SIN_HALLAZGO_TRANSITORIA	464	9.1%
ALUMBRADO_PUBLICO	339	6.6%
CANAL_INCORRECTO	295	5.8%
RECONEXION_ADMINISTRATIVA	209	4.1%
FACTURACION	151	3%
JURISDICCION_INCORRECTA	21	0.4%
TOTAL	**5.107**	**100%**

El análisis por tipología revela que las tres causas más frecuentes de visitas fallidas son: PODAS (23,5%), reportes sobre árboles en contacto con cables que se clasifican como falla eléctrica; DATO_INSUFICIENTE (18,8%), reportes con texto demasiado vago para determinar la naturaleza del problema; y FALLA_INTERNA_CLIENTE (17,6%), fallas en la instalación interna del cliente que no corresponden a la red de Celsia. Estas tres tipologías, junto con SERVICIO_AJENO (11,1%), concentran el 70,9% de los casos y constituyen las categorías prioritarias de detección del Agente Router.

4.4.3 Vectorización y construcción de ChromaDB

Una vez depurado y caracterizado el *dataset*, los textos normalizados fueron transformados en representaciones vectoriales (*embeddings*) mediante el modelo de lenguaje preentrenado paraphrase-multilingual-MiniLM-L12-v2 de *sentence-transformers*, que soporta español nativo y genera vectores de 384 dimensiones por texto.

Estrategia de compensación del desbalance

Para compensar el desbalance severo identificado en el EDA se aplicó *undersampling* estratificado sobre la categoría dominante. La siguiente tabla compara la distribución original del *dataset* con la distribución final en ChromaDB:

Tabla 21. Distribución original vs. muestreo estratificado para ChromaDB. El undersampling de SECTOR_SIN_ENERGIA reduce su representación del 91,2% al 38,9% en el espacio vectorial. Fuente: Elaboración propia.

Categoría	*Dataset* original	% original	Vectores ChromaDB	% en ChromaDB	Estrategia
SECTOR_SIN_ENERGIA	81.077	**91.2%**	**5.000**	**38.9%**	*Undersampling (seed=42)*
PROBLEMAS_VOLTAJE	4.308	4.8%	4.308	**33.5%**	*Todos los registros*
RIESGO_ANOMALIA	3.241	3.6%	3.235	**25.2%**	*Todos los registros*
SUMINISTRO_PARCIAL	285	0.3%	285	**2.2%**	*Todos los registros*
NO_ENERGETICO	33	0.04%	30	**0.2%**	*Todos los registros*
TOTAL	**88.944**	**100%**	**12.856**	**100%**	*Seed = 42*

El *undersampling* estratificado reduce la representación de SECTOR_SIN_ENERGIA del 91,2% al 38,9% en el espacio vectorial, dando mayor peso relativo a las categorías minoritarias. Esto es fundamental para que el clasificador pueda distinguir entre RIESGO_ANOMALIA (3,6% del *dataset* pero 25,2% de ChromaDB) y PROBLEMAS_VOLTAJE (4,8% pero 33,5% de ChromaDB), que de otro modo quedarían opacadas por la clase dominante.

Previo a adoptar el *undersampling*, se evaluaron dos alternativas. SMOTE fue descartado porque la interpolación lineal entre vectores semánticos no garantiza textos lingüísticamente coherentes en español, introduciendo ruido en el espacio vectorial y degradando la búsqueda por similitud coseno. La aumentación mediante *back-translation* fue descartada porque el lenguaje informal y fragmentado de los reportes produce traducciones de baja calidad que alteran el significado original, y el volumen de instancias minoritarias no justificaba el costo de validación manual requerido.

Se optó entonces por *undersampling* estratificado, reconociendo que implica descartar 76.077 de los 81.077 registros de la clase dominante. Esta pérdida de diversidad léxica se considera aceptable dado que el objetivo del clasificador es mejorar la discriminación entre clases, no maximizar la cobertura de la clase mayoritaria, y los 5.000 registros retenidos con *seed*=42 preservan la variabilidad representativa de esa categoría.

La elección de 5.000 como límite de muestreo para la clase dominante responde a un balance entre representatividad (suficiente diversidad de textos de esa categoría) y control del sesgo

(evitar que domine el espacio vectorial). La semilla aleatoria *seed*=42 garantiza la reproducibilidad de los resultados.

Proceso de vectorización

La vectorización se ejecutó mediante el script vectorizar.py con los siguientes pasos: (1) lectura del archivo EDA/dataset_final_limpio.csv; (2) eliminación de nulos; (3) mapeo de etiquetas REASON al esquema de categorías del sistema; (4) muestreo estratificado; (5) generación de *embeddings* por lotes de 256 registros con barra de progreso tqdm; (6) almacenamiento en ChromaDB con metadatos de categoría y etiqueta original.

El proceso procesó 12.856 registros en aproximadamente 3 minutos en un equipo estándar. El resultado es una base vectorial disponible en la carpeta ./chroma_db/ del proyecto, consultada en tiempo real por el Agente Clasificador mediante búsqueda de similitud coseno entre el vector del nuevo reporte y los 12.856 vectores almacenados.

Uso del *dataset* de visitas fallidas

Los 5.111 registros de visitas fallidas no fueron vectorizados en ChromaDB. En cambio, sus patrones lingüísticos fueron analizados manualmente para diseñar las reglas de detección del Agente Router: las palabras clave implementadas son reglas informadas por el EDA, interpretables y auditables, no aprendidas automáticamente por un modelo. Esta decisión garantiza que el comportamiento del Router sea predecible y fácilmente ajustable cuando se identifiquen nuevas tipologías de visitas fallidas en producción.

4.5 Selección y justificación de *frameworks* de orquesta conversacional

El desarrollo del sistema de agentes inteligentes propuesto requiere la elección de *frameworks* que permitan modelar flujos de decisión complejos, mantener estado persistente entre interacciones y definir roles especializados de agentes. Esta elección constituye una decisión arquitectónica de alto impacto: trabajar con el *framework* equivocado puede implicar reescrituras del 50% al 80% del código al escalar el sistema [34][35].

A continuación, se presenta una evaluación comparativa de los seis *frameworks* más relevantes del ecosistema actual (`LangGraph`, `CrewAI`, `AutoGen`, `Semantic Kernel`, `LlamaIndex` y `Haystack`), frente a los criterios de selección definidos según los requerimientos específicos del sistema de clasificación de reportes de daños de Celsia.

Los criterios de evaluación no fueron seleccionados de forma genérica sino derivados

directamente de los requisitos operativos identificados en la fase de empatía. El control de flujo con bifurcación condicional responde a la necesidad de rutas diferenciadas según tipo de falla y nivel de confianza. El estado persistente multiturno responde a que la conversación con el cliente puede requerir varios intercambios antes de clasificar el reporte. El soporte multi-agente nativo responde a la arquitectura de cuatro agentes especializados definida en la fase de ideación. La independencia de proveedor LLM responde al requisito de no generar dependencia adicional sobre la infraestructura Azure ya existente en Celsia. La madurez y adopción empresarial responde al requisito de viabilidad de implementación en un entorno productivo regulado. A estos criterios se suma el costo estimado por conversación, relevante para evaluar la sostenibilidad económica del sistema a escala del volumen de LuzIA (83.707 interacciones mensuales).

Tabla 22. Comparativa de frameworks de orquestación de agentes inteligentes

Criterio de evaluación	***LangGraph***	***CrewAI***	***AutoGen (Microsoft)***	***Semantic Kernel (Microsoft)***	***LlamaIndex / Haystack***
Arquitectura central	Grafo de estado dirigido (StateGraph)	Roles y tareas jerárquicas (Crew)	Conversación multi-agente (chat grupal)	Plugins y planificador semántico	RAG / pipelines de recuperación
Control de flujo	Alto: ciclos, aristas condicionales, ejecución paralela	Medio-alto: secuencial, paralelo, jerárquico	Dinámico por conversación, menos predecible	Alto: planificador automático	Bajo-medio: orientado a recuperación
Estado persistente	Centralizado en `StateGraph (TypedDict)`	Por agente y *crew*	Historial conversacional acumulado	Por *plugin* / sesión	Índice vectorial
Soporte multi-agente nativo	Sí (subgrafos y nodos paralelos)	Sí (crews con roles definidos)	Sí (conversaciones grupales)	Parcial (vía orquestador SK)	No nativo
Independencia de proveedor LLM	Alta (OpenAI, Anthropic, local)	Alta (agnóstico)	Media (GPT-centric)	Baja (Azure/OpenAI	Media-alta

				preferente)	
Curva de aprendizaje	Alta (requiere conocimiento de grafos)	Media (abstracción de roles intuitiva)	Media	Media-alta	Media
Madurez y adopción	GA mayo 2025 · +400 empresas en prod.	Producción 2024-2025 · 60% Fortune 500	Fusionado con SK (oct. 2025), en preview	GA en .NET, Python en mejora continua	Producción, especializado en RAG
Mejor caso de uso	Flujos con bifurcación, estado explícito y ciclos	Equipos de agentes con roles especializados	Colaboración conversacional y prototipos rápidos	Integración en ecosistema .NET / Azure	Agentes con RAG intensivo sobre documentos
Costo estimado por conversación	~$0.00006 USD por conversación (*embeddings* locales sin costo adicional + LLM juez activado ~30% de casos a ~500 tokens promedio)	Variable según LLM configurado	Variable según modelo GPT utilizado	Incluido en suscripción Azure si ya contratado	Bajo (modelos locales o *embeddings* abiertos)
Adecuación al proyecto	★★★★★	★★★★★	★★★☆☆	★★☆☆☆	★★☆☆☆

Fuente: elaboración propia a partir de [34][35][36][37][42][43][A17][A18].

Justificación de la selección de LangGraph y CrewAI

La elección de `LangGraph y CrewAI` como *frameworks* para el presente proyecto responde a cinco criterios definidos según los requerimientos específicos del sistema:

- **Control de flujo con bifurcación condicional:** El proceso de clasificación requiere rutas diferenciadas según el tipo de falla detectada (energético versus no energético) y niveles de confianza variables. `LangGraph` provee aristas condicionales que implementan esta lógica de manera explícita, auditable y sin depender de instrucciones en los *prompts* del LLM, garantizando comportamiento determinista [37][38]. En el prototipo implementado,

esta capacidad se refleja en las cinco aristas condicionales del `StateGraph` que dirigen el flujo entre los cuatro agentes según el valor de tipo_falla, score_confianza e intentos_validacion.

- **Estado persistente multiturno:** La conversación con el cliente puede requerir múltiples intercambios antes de obtener información suficiente para clasificar el reporte. El `StateGraph` de `LangGraph` mantiene el historial completo de la conversación y los resultados intermedios de cada agente sin necesidad de gestión manual del contexto entre turnos [37][40].
- **Especialización por roles:** El sistema requiere agentes con responsabilidades claramente delimitadas: clasificación semántica, validación de coherencia, enrutamiento y creación del evento. La arquitectura de roles de `CrewAI`, definida mediante los *atributos role, goal y backstory* de cada agente, facilita esta especialización y el traspaso estructurado de contexto entre agentes, sin solapamiento de responsabilidades [42][43].
- **Independencia de ecosistema**: Se priorizaron *frameworks* agnósticos al proveedor de LLM y de nube. Tanto *`LangGraph`* como `CrewAI` son compatibles con múltiples proveedores de modelos (OpenAI, Anthropic, modelos locales), sin generar dependencia con un único vendor [35][A16]. En el prototipo, esta independencia se evidencia en el uso de Claude Haiku (Anthropic)/ GPT-4o-mini (OpenAI) como LLM juez en la Capa 2 del Clasificador, intercambiable por cualquier otro modelo sin modificar la arquitectura del grafo.
- **Madurez y adopción empresarial:** Ambos *frameworks* cuentan con disponibilidad general confirmada y casos de uso documentados en producción a escala empresarial: `LangGraph` opera en más de 400 empresas incluyendo LinkedIn y Uber; `CrewAI` está presente en el 60% de las empresas Fortune 500 [35][A16].

Razones de descarte de las demás alternativas

- **AutoGen**: Fue descartado por carecer de estado persistente explícito entre turnos y por su enfoque conversacional libre, que dificulta la implementación de flujos deterministas con umbrales de confianza definidos. Adicionalmente, en octubre de 2025 fue fusionado con Semantic Kernel en el Microsoft Agent Framework, que aún se encuentra en public preview con fecha de disponibilidad general prevista para el primer trimestre de 2026 [35].
- **Semantic Kernel**: Fue descartado por su dependencia primaria del ecosistema .NET y Azure, con paridad incompleta en Python (el lenguaje del proyecto), y por el riesgo de *vendor lock-in* asociado a la integración con servicios de Microsoft [A17][35].
- **LlamaIndex:** Fue descartado por estar orientado principalmente a recuperación de documentos (RAG intensivo) y no a flujos de clasificación con lógica de decisión compleja y múltiples agentes especializados con roles diferenciados [A18][35].

- **Haystack**: Fue descartado porque su arquitectura de pipelines está diseñada principalmente para búsqueda semántica y respuesta a preguntas sobre documentos, no para orquestación multi-agente con estado conversacional persistente y ciclos de validación iterativa [A18].

La combinación de `LangGraph` como orquestador y `CrewAI` como *framework* de agentes responde al principio de responsabilidad única: `LangGraph` gestiona el estado, el control de flujo y los ciclos de validación; `CrewAI` gestiona la especialización, los roles y el traspaso de contexto entre agentes. Esta separación de responsabilidades facilita el mantenimiento, la depuración y la escalabilidad del sistema.

4.6 Stack tecnológico implementado

El prototipo fue construido íntegramente en Python 3.12 con las siguientes librerías y *frameworks*:

Tabla 23. Stack tecnológico del prototipo funcional implementado. Fuente: Elaboración propia.

LIBRERÍA / FRAMEWORK	ROL EN EL SISTEMA	INSTALACIÓN
Python 3.12	Lenguaje base del sistema	-
`LangGraph`	Orquestador central (`StateGraph` + aristas condicionales)	`pip install langgraph`
`CrewAI`	Framework de agentes especializados *(role, goal, backstory)*	`pip install crewai`
sentence-transformers	Embeddings multilingüe (paraphrase-multilingual-MiniLM-L12-v2)	`pip install sentence-transformers`
ChromaDB	Base de datos vectorial local (12.856 vectores del *dataset* Celsia)	`pip install chromadb`
Anthropic SDK	LLM juez (Claude Haiku) vía API - segunda capa del clasificador	`pip install anthropic`
OpenAI SDK	LLM juez (GPT-4o-mini) vía API	`pip install openai`
FastAPI + Uvicorn	Servidor web - expone el sistema como API REST	`pip install fastapi uvicorn`
python-telegram-bot	Bot de Telegram (@Luzia_celsia_bot) - canal de pruebas	`pip install python-telegram-bot`
SQLite (built-in)	Persistencia de reportes - dashboard y generación de informes	Nativo en Python
tqdm + pandas	Vectorización por lotes con barra de progreso	`pip install tqdm pandas`

4.7 Implementación del prototipo

El sistema fue desarrollado íntegramente en Python 3.12 y desplegado sobre dos canales simultáneos: una interfaz web construida con FastAPI y un bot público de Telegram (@Luzia_celsia_bot), ambos conectados al mismo grafo de agentes y compartiendo la misma base de datos de persistencia.

Paso 1 - Preparación del dataset vectorizado

Con el *dataset* depurado (sección 4.4), se generaron los *embeddings* utilizando el modelo paraphrase-multilingual-MiniLM-L12-v2 y se almacenaron en ChromaDB. La vectorización se ejecutó en aproximadamente 3 minutos en un equipo estándar, procesando 12.856 registros en lotes de 256.

Paso 2 - Construcción de los agentes con CrewAI

Cada agente fue implementado como una función Python que recibe el `TypedDict` EstadoReporte y devuelve un diccionario con los campos actualizados. La integración con `CrewAI` se realizó a nivel de roles y lógica de especialización; la orquestación entre agentes se delegó completamente a `LangGraph`.

Paso 3 - Orquestación con LangGraph

Se definió el `StateGraph` con cinco nodos (clasificador, router, validador, creador_evento, fallback) y aristas condicionales que dirigen el flujo según el tipo_falla y la decision_evento. El ciclo del Validador se implementa como una arista de retorno del nodo validador a sí mismo, controlada por el campo intentos_validacion.

Paso 4 - Exposición como API con FastAPI

El grafo compilado se expone mediante tres *endpoints*: *POST* /procesar (ejecuta el flujo completo de agentes), *GET /stats* (devuelve estadísticas en tiempo real desde SQLite) y *GET /dashboard* (interfaz de monitoreo). La interfaz web replica el flujo del sistema actual de Celsia con un menú de motivos como primer filtro, *popups* de redirección para reportes no energéticos y cierre explícito del flujo tras cada decisión.

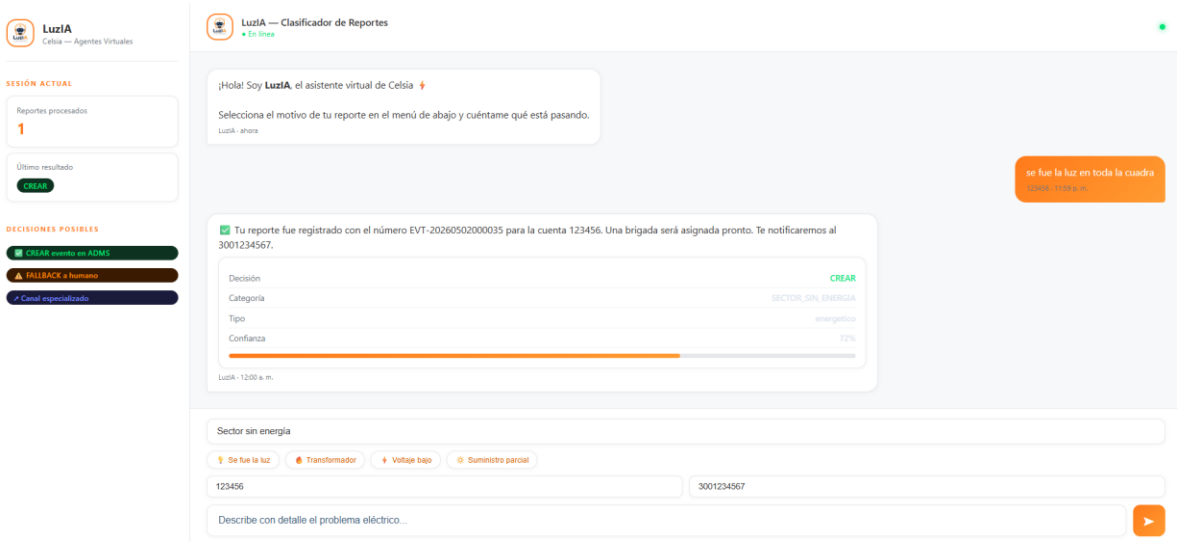


Ilustración 8. Interfaz Web para reporte de daños

Paso 5- Conexión a Telegram para pruebas

El bot @Luzia_celsia_bot fue creado mediante @BotFather y conectado al grafo mediante python-telegram-bot. El flujo conversacional en Telegram replica el de la interfaz web: menú de motivos como primer mensaje, seguido de código de cuenta, celular y descripción libre. Los reportes no energéticos reciben respuesta inmediata sin pasar por los agentes. Todos los reportes procesados por ambos canales quedan registrados en SQLite con su canal de origen, código de cuenta y usuario de Telegram cuando aplica. En la Ilustración 9 se muestra cómo fue creado el bot Luzia_celsia_bot a través de BotFather en telegram y la Ilustración 10 muestra el chatbot en ejecución.

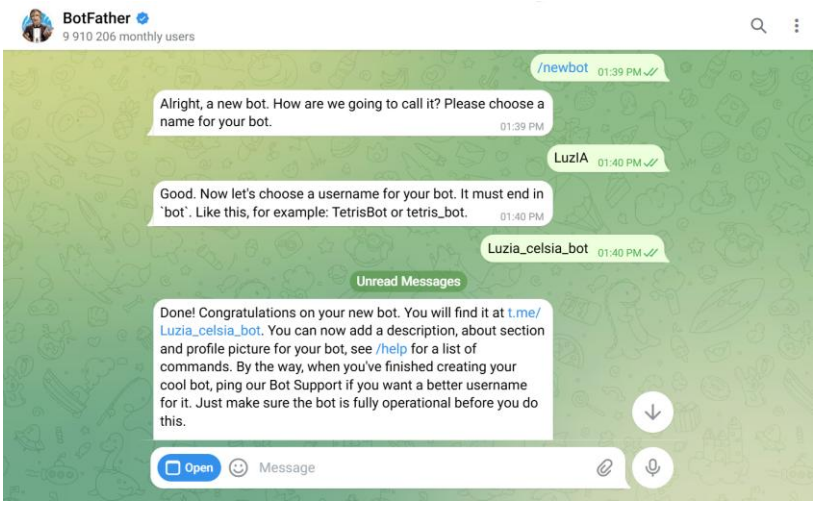


Ilustración 9. Creación del bot en BotFather de telegram

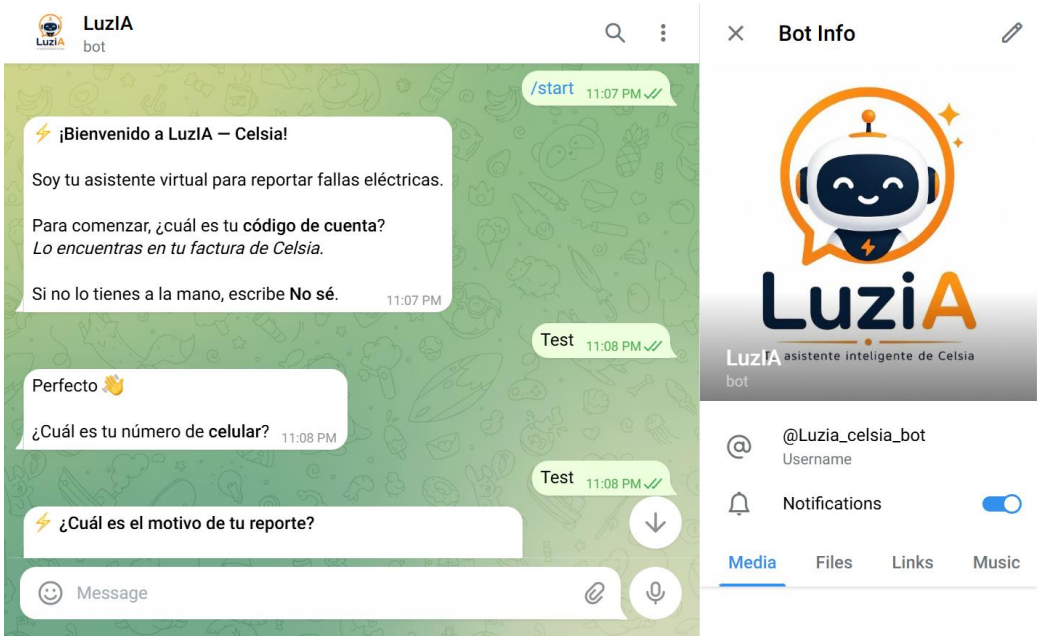


Ilustración 10. Bot @Luzia_celsia_bot de telegram

Paso 6 - Implementación del dashboard de monitoreo

Como componente complementario al prototipo, se desarrolló un *dashboard* de monitoreo accesible desde el *endpoint* */dashboard*, el cual permite visualizar en tiempo real el comportamiento del sistema a partir de los reportes procesados y almacenados en la base de datos SQLite.

El *dashboard* consolida información proveniente de ambos canales (interfaz web y Telegram), permitiendo analizar indicadores clave como: número total de reportes, score promedio de clasificación, decisiones tomadas por el sistema (creación automática o *fallback* humano), y distribución de reportes por canal de origen.

Adicionalmente, el *dashboard* incluye métricas específicas orientadas al objetivo del proyecto, como la identificación de tipologías de visitas fallidas evitadas (por ejemplo, podas, servicio ajeno o datos insuficientes), así como el funcionamiento interno del sistema de agentes, mostrando la activación de cada capa del clasificador (*embeddings* y LLM). Esta herramienta no solo permite validar el desempeño del sistema en un entorno controlado, sino que también constituye un mecanismo de trazabilidad y análisis para futuras fases de implementación en producción.

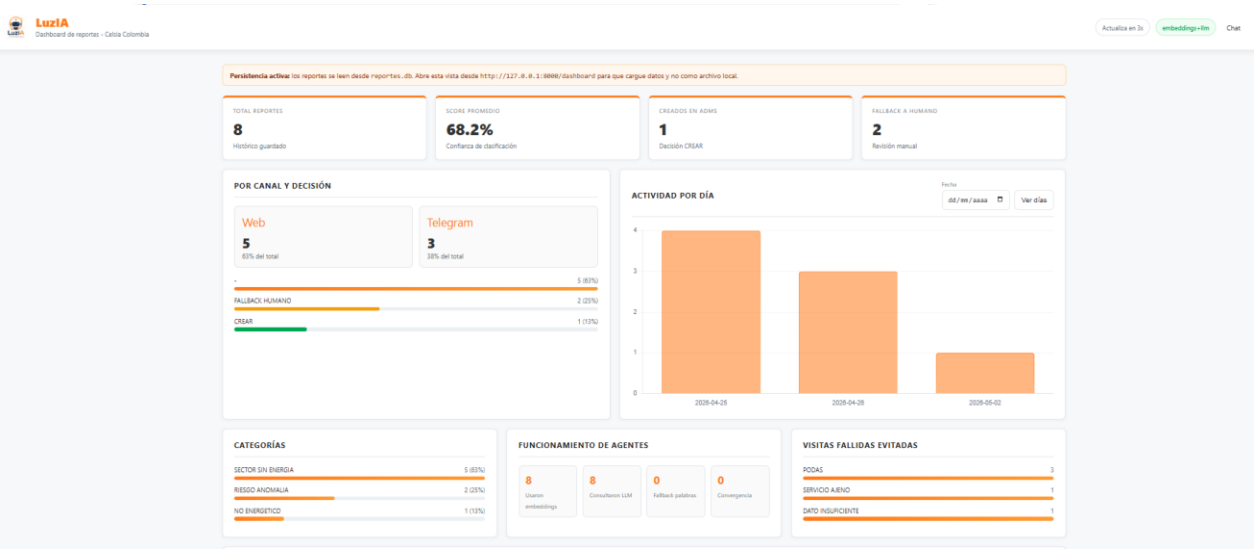


Ilustración 11. Parte 1 Dashboard de monitoreo

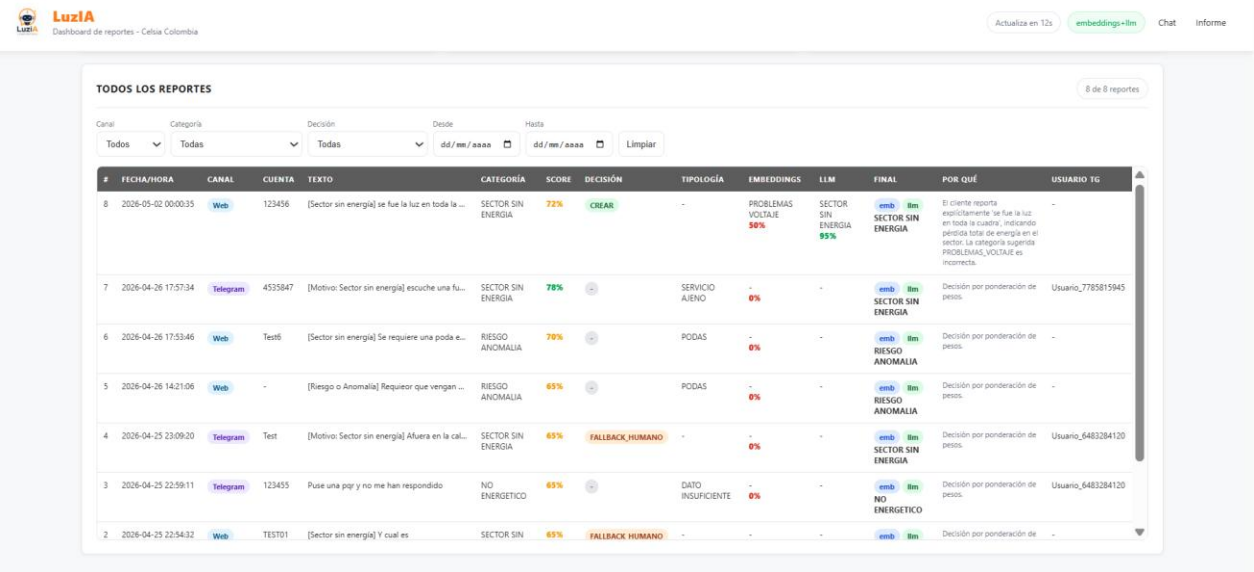


Ilustración 12. Parte 2 Dashboard de monitoreo

En conjunto, la arquitectura propuesta, el diseño de los agentes, el mecanismo de clasificación híbrido y el procesamiento del *dataset* permiten consolidar un sistema funcional capaz de interpretar, validar y gestionar reportes de incidentes en un entorno conversacional real. Más allá de la implementación técnica, este capítulo evidencia cómo las decisiones de diseño adoptadas responden directamente a las problemáticas identificadas en el análisis exploratorio, particularmente la brecha semántica en la clasificación y la generación de visitas técnicas fallidas.

Sobre esta base, el sistema se encuentra en condiciones de ser evaluado en términos de desempeño, precisión y confiabilidad. En este sentido, el siguiente capítulo presenta los resultados de la validación del agente virtual en el canal de atención al cliente, en cumplimiento del Objetivo Específico 3.

5. VALIDACIÓN DE AGENTE VIRTUAL Y EVALUACIÓN DE DESEMPEÑO OPERATIVO

El presente capítulo tiene como objetivo evaluar el desempeño técnico, operativo y funcional del agente virtual inteligente ("LuzIA") desarrollado para la clasificación automática de reportes de incidentes en los canales digitales de Celsia Colombia con el modelo de Claude y el modelo de OpenAI. A partir de las brechas de clasificación identificadas previamente, se estableció un marco de validación riguroso que permite contrastar la efectividad de la arquitectura híbrida propuesta (*Embeddings* y Modelos de Lenguaje Grande - LLM) frente al modelo operativo base. La evaluación se estructura mediante un diseño experimental aplicado a casos históricos reales, un análisis exhaustivo de métricas de clasificación multiclase, la proyección de la reducción de visitas técnicas fallidas, pruebas de consistencia mediante validación cruzada y el protocolo para la evaluación cualitativa de usabilidad del sistema.

5.1 Diseño Experimental

Para garantizar que la validación del sistema reflejara fielmente las condiciones operativas y la complejidad semántica de las interacciones reales con los clientes, el diseño experimental se estructuró bajo una **muestra mixta y balanceada** de 1.000 casos reales extraídos del historial de atención de Celsia. El diseño experimental se estructuró de la siguiente manera:

- **500 visitas fallidas confirmadas**, etiquetadas bajo la categoría objetivo "NO_ENERGETICO" (NE).
- **500 reportes energéticos legítimos,** balanceados equitativamente en 125 casos para cada una de las cuatro tipologías operativas (Sector Sin Energía, Problemas de Voltaje, Riesgo/Anomalía y Suministro Parcial). Para evitar solapamientos y sesgos, se excluyeron 5.030 identificadores (GID) de visitas fallidas del muestreo energético.

La selección de esta muestra se realizó mediante un muestreo aleatorio controlado utilizando una semilla de reproducibilidad fijada en 42 (*seed*=42). Adicionalmente, esta iteración incorporó la ejecución de reglas directamente dentro del clasificador y un "*prompt*" optimizado. Esto resulta fundamental para validar que el sistema no solo filtra las solicitudes ambiguas (visitas fallidas), sino que también protege activamente los reportes reales de fallas energéticas, evitando despachos erróneos sin afectar la calidad del servicio.

El experimento se dividió en tres escenarios de prueba, los cuales varían estructuralmente en el grado de predominancia (peso algorítmico) asignado a la capa de similitud vectorial (*Embeddings* - E) frente a la capa de razonamiento del LLM. Estos escenarios permiten determinar empíricamente la configuración óptima para alcanzar el mayor nivel de asertividad.

5.2 Evaluación por Escenarios y Métricas Principales modelo (Claude)

Para cada uno de los tres escenarios diseñados se midieron indicadores de rendimiento robustos, incluyendo Exactitud (*Accuracy*), Sensibilidad (*Recall*), F1-Score Macro, así como la retención de incidentes energéticos (es decir, evitar falsos negativos en reportes reales). Las categorías operativas evaluadas fueron: Sector Sin Energía (SSE), Problemas de Voltaje (PV), Riesgo/Anomalía (RA), Suministro Parcial (SP) y No Energético (NE).

5.2.1 Escenario 1: Predominancia *Embeddings* (E:0.7 / LLM:0.3)

En el primer escenario, se otorgó un peso del 70% a la similitud vectorial basada en *Embeddings* y un 30% al juez LLM. Bajo esta configuración, el sistema evidenció un fallo crítico en la comprensión de contextos semánticos ambiguos. El rendimiento general arrojó un ***Accuracy* de apenas 32.4%** y un F1 Macro de 0.3027.

La capacidad del sistema para identificar reportes ajenos a daños eléctricos (Recall NE) fue muy deficiente, alcanzando solo el 25.2% (apenas 126 de 500 casos). Sin embargo, al tener un comportamiento tan conservador, este modelo retuvo erróneamente casi todo en las categorías técnicas, lo que resultó en una Retención Energética del 96.8% (solo 16 falsos negativos energéticos). Su incapacidad para detectar el "ruido" de las visitas fallidas descarta este escenario como una solución operativa viable.

5.2.2 Escenario 2: Predominancia LLM (E:0.3 / LLM:0.7) - Configuración Óptima

El segundo escenario invirtió los pesos algorítmicos, otorgándole al razonamiento deductivo del LLM una predominancia del 70%, mientras que los Embeddings aportaron el 30%. Los resultados obtenidos bajo esta arquitectura demostraron una mejora sustancial en la capacidad predictiva, consolidándose como la configuración más asertiva del estudio.

Esta iteración alcanzó el **mejor *Accuracy* global con un 64.0%** y un F1 Macro de 0.5953. El sistema logró detectar de forma exitosa el **65.8% de las visitas fallidas reales** (clase "NO_ENERGETICO").

A la par de este incremento en la detección, el modelo logró un desempeño sobresaliente en la protección del flujo técnico: obtuvo una **Retención Energética del 93.6%**. Esto significa que de los 500 reportes que sí eran fallas reales en la red, el sistema solo confundió 32 (Falsos Negativos), logrando clasificar adecuadamente el 68.8% de los Sector Sin Energía (SSE) y el 75.2% de los Riesgos/Anomalías (RA). Por su equilibrio entre filtrado inteligente y seguridad operativa, este escenario se cataloga unánimemente como el "Mejor Escenario" y constituye la recomendación final.

5.2.3 Escenario 3: Pesos Iguales (E:0.5 / LLM:0.5)

Con el fin de agotar las posibilidades de diseño, se evaluó un tercer escenario asignando un 50% de responsabilidad a cada capa tecnológica. Los resultados de este enfoque equilibrado fueron consistentes, pero mostraron valores prácticamente idénticos a los del Escenario 2.

El sistema obtuvo un *Accuracy* de 63.9%, un F1 Macro de 0.5932 y una detección del 65.8% en los casos no energéticos reales (NO_ENERGETICO), junto a una Retención Energética del 93.8% (31 Falsos Negativos). Aunque es un modelo robusto, la leve superioridad matemática del Escenario 2 en F1-Score reafirma la necesidad de darle predominancia al razonamiento profundo del LLM sobre la búsqueda vectorial.

5.3 Evaluación por Escenarios y Métricas Principales modelo (OpenAI)

Para cada uno de los tres escenarios diseñados se midieron indicadores de rendimiento robustos, incluyendo Exactitud (*Accuracy*), Sensibilidad (*Recall*), F1-Score Macro, así como la retención de incidentes energéticos (es decir, evitar falsos negativos en reportes reales). Las categorías operativas evaluadas fueron: Sector Sin Energía (SSE), Problemas de Voltaje (PV), Riesgo/Anomalía (RA), Suministro Parcial (SP) y No Energético (NE).

5.3.1 Escenario 1: Predominancia Embeddings (E:0.7 / LLM:0.3)

En este escenario inicial, se otorgó un peso predominante al motor de búsqueda vectorial. El rendimiento general arrojó un *Accuracy* de 32.4% y un F1 Macro de 0.3015. La capacidad para identificar reportes ajenos a daños eléctricos (*Recall* NE) fue limitada, alcanzando solo el 25.2% (126 de 500 casos). Sin embargo, mostró una alta Retención Energética del 96.8% (solo 16 falsos negativos), confirmando que la búsqueda vectorial tiende a ser extremadamente conservadora.

5.3.1 Escenario 2: Predominancia LLM (E:0.3 / LLM:0.7)

Al otorgarle al razonamiento deductivo de GPT-4o-mini una predominancia del 70%, el rendimiento mejoró sustancialmente. Este escenario alcanzó un *Accuracy* de 58.6% (frente al 64.0% de Claude) y un F1 Macro de 0.5623. La detección de visitas fallidas (*Recall* NE) subió al 53.4% (muy por debajo del 65.8% de Claude). La Retención Energética se situó en 96.6% (17 falsos negativos), demostrando un salto cualitativo en la comprensión de la intención del cliente.

5.3.2 Escenario 3: Predominancia LLM (E:0.5 / LLM:0.5)

Mejor configuración interna para OpenAI. A diferencia del modelo de Claude, el punto de equilibrio óptimo para OpenAI se encontró en la asignación equitativa de pesos. El sistema obtuvo

un *Accuracy* de 58.6% y un F1 Macro de 0.5620. Aunque el *Accuracy* es idéntico al Escenario 2, el Escenario 3 logró una detección ligeramente superior de casos no energéticos (53.6% / 268 casos), manteniendo la misma Retención Energética de 96.6% (17 falsos negativos). Esta configuración se selecciona como la óptima para este modelo por su balance entre seguridad operativa y filtrado de ruido. Aunque es el "mejor escenario" dentro de las pruebas de OpenAI, sus métricas globales y de detección de visitas fallidas son inferiores a las de Claude.

5.4 Análisis de Visitas Fallidas y Detección por Tipología para ambos modelos

El núcleo de valor operativo de este proyecto radica en la prevención de desplazamientos injustificados de las brigadas técnicas de Celsia hacia terrenos donde no existen fallas de la infraestructura eléctrica. Con el metodológico de evaluación y la configuración óptima (Escenario 2), el modelo de Claude alcanzó una tasa de detección real del 65.8% sobre el ruido operativo, mitigando sustancialmente las visitas fallidas.

Para el modelo de Claude, el desglose de la detección del clasificador por tipologías operativas revela que el sistema es altamente efectivo discriminando solicitudes comerciales, informativas o de fallas evidentes no atribuibles a Celsia. Las tasas de detección (*Recall*) logradas para las diferentes causas raíz fueron las siguientes:

Tabla 24. Detección NO_ENERGETICO por Tipología para el modelo de Claude.

Tipología Operativa	Total de Casos	Detectados (Filtrados)	Recall (Efectividad)
FACTURACION	15	15	100.0%
RECONEXION_ADMINISTRATIVA	22	21	95.5%
CANAL_INCORRECTO	31	29	93.5%
ALUMBRADO_PUBLICO	32	29	90.6%
SERVICIO_AJENO	56	50	89.3%
SIN_HALLAZGO_TRANSITORIA	32	27	84.4%
FALLA_INTERNA_CLIENTE	77	58	75.3%
DATO_INSUFICIENTE	100	72	72.0%
PODAS	130	28	21.5%
JURISDICCION_INCORRECTA	5	0	0.0%

El análisis demuestra que, en 8 de las 10 tipologías, el sistema supera la barrera del 70% de asertividad. La principal limitante algorítmico, y la razón por la cual el modelo no supera un *Accuracy* global mayor, es la fricción analítica con la categoría de **Podas**. La ambigüedad lingüística entre un usuario reportando "ramas rozando cables" y otro reportando "ramas que reventaron un cable y dejaron sin energía", sin soporte visual o fotográfico, representa un desafío semántico

que el LLM aún clasifica conservadoramente como posibles emergencias eléctricas.

Ahora bien, bajo su mejor configuración (Escenario 3), el modelo de OpenAI alcanzó una tasa de detección real de apenas el **53.6%** sobre la clase NO_ENERGETICO. A diferencia de la capacidad analítica demostrada por Claude (**65.8%**), GPT-4o-mini exhibe un comportamiento conservador frente a la ambigüedad semántica.

Para el modelo de OpenAI, el desglose de la detección del clasificador por tipologías operativas revela que el sistema es menos efectivo discriminando solicitudes comerciales, informativas o de fallas evidentes no atribuibles a Celsia. Las tasas de detección (*Recall*) logradas para las diferentes causas raíz fueron las siguientes:

Tabla 25. Detección NO_ENERGETICO por Tipología para el modelo de OpenAI.

Tipología Operativa	Total de Casos	Detectados (Filtrados)	Recall (Efectividad)
FACTURACION	15	15	100.0%
SERVICIO_AJENO	56	49	87.5%
RECONEXION_ADMINISTRATIVA	22	19	86.4%
ALUMBRADO_PUBLICO	32	27	84.4%
CANAL_INCORRECTO	31	26	83.9%
FALLA_INTERNA_CLIENTE	77	51	66.2%
SIN_HALLAZGO_TRANSITORIA	32	19	59.4%
DATO_INSUFICIENTE	100	35	35.0%
PODAS	130	27	20.8%
JURISDICCION_INCORRECTA	5	0	0.0%

Si bien este conservadurismo extremo reduce los falsos negativos energéticos a un 3.4%, penaliza gravemente el objetivo primario de la tesis: la reducción de visitas fallidas. Al dejar pasar casi la mitad del "ruido operativo" hacia el sistema de cuadrillas, OpenAI no logra cumplir con las expectativas operativas del negocio, justificando su descarte frente a Claude.

5.5 Análisis de costos de API

Más allá de la eficiencia técnica y la precisión en la clasificación, la viabilidad de implementar un sistema basado en LLM en un entorno de producción real está condicionada por su sostenibilidad económica. En este apartado se presenta una comparativa financiera detallada, tomando como referencia los precios de mercado vigentes a mayo de 2026.

Tal como se detalla en la **Tabla 6.1**, existe una disparidad significativa entre los esquemas de

precios de los proveedores evaluados. Mientras que el modelo GPT-4o-mini de OpenAI se posiciona como la opción más económica con un costo de $0.15 USD por millón de tokens de entrada, Claude Haiku 4.5 de Anthropic presenta una tarifa de $1.00 USD para el mismo volumen, lo que representa una estructura de costos 7.2 veces superior. Esta diferencia se refleja directamente en la fase de evaluación V3, donde el procesamiento de los escenarios de prueba supuso un gasto de $3.45 USD para el modelo de Anthropic frente a apenas $0.48 USD para la solución de OpenAI.

Tabla 26. Precios oficiales por millón de tokens (mayo 2026).

Modelo	*Input* ($/M tok)	*Output* ($/M tok)	Costo Evaluación	Relación de Precio
Claude Haiku 4.5	$1.00	$5.00	**$3.45 USD**	7.2x más caro
GPT-4o-mini	$0.15	$0.60	**$0.48 USD**	Referencia

Para contextualizar estos valores en la operación diaria de la compañía, se realizó una proyección basada en el volumen histórico de 86,000 reportes anuales procesados por Celsia. Los resultados de esta simulación, presentados en la **Tabla 6.2**, permiten identificar dos escenarios de despliegue claramente diferenciados. Bajo una tarifa estándar de procesamiento en tiempo real, el uso de GPT-4o-mini permitiría gestionar la totalidad de la demanda anual con una inversión de tan solo $13.80 USD, lo que supone un ahorro directo de $85.10 USD en comparación con el despliegue bajo Claude Haiku 4.5.

Tabla 27. Proyección de costos en producción (86.000 reportes/año).

Escenario de Procesamiento	Claude Haiku 4.5	GPT-4o-mini	Ahorro con OpenAI
Tarifa Estándar (Tiempo real)	$98.90 USD/año	$13.80 USD/año	**$85.10 USD**
Con Batch API (50% descuento)	$49.45 USD/año	$6.90 USD/año	**$42.55 USD**

Finalmente, es importante destacar el potencial de optimización mediante el uso de *Batch* APIs. Como se observa en la misma tabla de proyecciones, la adopción de procesamiento por lotes, que permite un descuento del 50% en las tarifas, reduciría el costo operativo anual de la solución de OpenAI a una cifra marginal de $6.90 USD. Estos datos sugieren que, desde una perspectiva estrictamente financiera, el modelo GPT-4o-mini ofrece una escalabilidad superior, permitiendo la automatización de la clasificación de reportes con un impacto económico prácticamente despreciable para la organización.

5.6 Impacto de las Reglas Operativas y Robustez del Modelo

Para garantizar la robustez operativa del sistema, el diseño metodológico implementó una

estructura basada en el balanceo de los datos de entrada y la intervención guiada mediante reglas en el clasificador.

La inclusión de una muestra mixta (50% fallas reales, 50% visitas fallidas) certificó que el modelo está estadísticamente equilibrado y no categoriza las solicitudes sesgadamente en una sola dirección. Sumado a esto, se incorporaron reglas de negocio codificadas, como la "*regla_ne_alta_precision"*, que se ejecutó 42 veces de forma certera. Esta sinergia entre inteligencia generativa probabilística y reglas determinísticas garantizó que la **tasa de Falsos Negativos (reportes reales de energía descartados) se limitara a un ínfimo 6.4%** (32 de 500). Esto avala que el modelo de Claude es seguro para salir a producción, ya que minimiza el riesgo de desatención de emergencias reales.

Por otro lado, la ejecución de reglas determinísticas dentro del clasificador de OpenAI, como la regla_ne_alta_precision, se activó correctamente para forzar la clasificación de casos evidentes. Sin embargo, ni siquiera el apoyo de estas reglas fue suficiente para compensar las deficiencias analíticas del LLM de OpenAI a la hora de desambiguar el contexto real del cliente, reafirmando la recomendación técnica de implementar el sistema sobre Claude Haiku.

5.7 Evaluación de Usabilidad - *System Usability Scale* (SUS)

Dado que un componente vital de los sistemas conversacionales es la percepción, la facilidad de uso y la adopción por parte de quienes interactúan con él, el diseño de la evaluación incorpora el estándar de medición de usabilidad *System Usability Scale* (SUS). Actualmente, esta fase se encuentra en proceso de coordinación con el equipo operativo de Celsia Colombia, habiéndose programado su ejecución práctica para mayo de 2026.

El protocolo establecido seleccionará a 12 operadores especialistas en atención al cliente, quienes someterán al sistema a sesiones de 30 minutos. Durante estas sesiones de prueba, cada analista deberá ejecutar cuatro tareas críticas en la herramienta: clasificar un reporte de sector sin energía, clasificar un reporte de poda de árboles, clasificar un reporte de daño en poste, y la tarea más compleja, interpretar las repuestas guiadas del sistema ante el ingreso de un reporte deliberadamente ambiguo.

Posterior a la ejecución, se diligenciará el cuestionario SUS compuesto por 10 ítems estándar que intercalan afirmaciones positivas (p. ej., "Creo que me gustaría usar este sistema con frecuencia", "Pensé que el sistema era fácil de usar", "Me imagino que la mayoría de las personas aprenderían a usar este sistema muy rápidamente", "Me sentí muy seguro/a al usar el sistema") y negativas (p. ej., "Encontré el sistema innecesariamente complejo", "Encontré el sistema muy difícil de usar", "Creo que había demasiada inconsistencia en este sistema").

Las respuestas se recogerán bajo una escala de Likert de 1 a 5 (donde 1 es totalmente en desacuerdo y 5 totalmente de acuerdo) y se procesarán utilizando la fórmula convencional del instrumento: $SUS = 2.5 \times [\Sigma(\text{positivas} - 1) + \Sigma(5 - \text{negativas})]$

El cálculo del puntaje final se basa en la transformación de sus componentes individuales según la naturaleza de cada pregunta. En primer lugar, para los ítems con numeración impar, que corresponden a las afirmaciones positivas (1, 3, 5, 7 y 9), se toma la puntuación otorgada por el usuario y se le resta una unidad. Por el contrario, en las afirmaciones con numeración par o negativas (2, 4, 6, 8 y 10), el valor se determina restando la calificación del usuario al número cinco. Finalmente, se aplica un multiplicador de 2.5 a la suma de estos resultados parciales, lo cual permite normalizar la cifra final en una escala estándar de 0 a 100.

El objetivo operativo pactado (*benchmark*) es que la herramienta alcance una puntuación igual o superior a 68, lo cual sitúa al diseño propuesto estadísticamente por encima del promedio del sector eléctrico, categorizando su usabilidad entre "Sobre el promedio" o "Excelente" (Grados C, B o A). Los resultados definitivos de esta escala serán incorporados oportunamente al documento final.

5.8 Resultados de la Evaluación de usabilidad

Durante el mes de mayo de 2026, se llevó a cabo una evaluación detallada del agente virtual conocido como **@Luzia_celsia_bot**, el cual opera a través de la plataforma Telegram. Para este proceso, se contó con una muestra representativa de 12 operadores y usuarios, quienes participaron activamente en la medición de la herramienta.

Escala de medición

Para esta evaluación se aplicó el cuestionario estandarizado de 10 preguntas en una escala de Likert del 1 al 5.

Tabla 28. Preguntas de evaluación de la usabilidad.

No.	Descripción de la Pregunta	Tipo
P1	Me gustaría usar este sistema con frecuencia.	(+)
P2	Encontré el sistema innecesariamente complejo.	(-)
P3	Pensé que el sistema era fácil de usar.	(+)
P4	Necesitaría apoyo técnico para usar este sistema.	(-)
P5	Las diversas funciones estaban bien integradas.	(+)
P6	Pensé que había demasiada inconsistencia.	(-)
P7	La mayoría aprendería a usarlo muy rápidamente.	(+)
P8	Encontré que el sistema era muy engorroso de usar.	(-)

P9	Me sentí muy seguro/a al usar el sistema.	(+)
P10	Necesitaba aprender muchas cosas antes de empezar.	(-)

A continuación, se desglosan las puntuaciones individuales otorgadas por cada uno de los 12 evaluadores que formaron parte de la muestra. La tabla 24 presenta la valoración asignada a cada uno de los diez ítems del cuestionario SUS, permitiendo observar la variabilidad de las respuestas y el cálculo específico de la suma parcial para cada caso. Estos datos crudos constituyen la base fundamental para el cálculo del score final, facilitando la identificación de patrones de satisfacción o dificultades puntuales encontradas durante la interacción con el agente virtual.

Tabla 29. Resultados Detallados por Usuario

Usuario	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	Score Final
Usuario 1	5	2	5	1	4	1	5	1	4	1	**92.5**
Usuario 2	4	2	4	2	4	2	4	2	4	2	**75.0**
Usuario 3	5	1	5	1	5	1	5	1	5	1	**100.0**
Usuario 4	4	3	4	2	4	2	4	2	3	2	**70.0**
Usuario 5	5	2	4	1	4	2	5	1	4	1	**87.5**
Usuario 6	4	1	5	1	4	1	5	1	4	2	**90.0**
Usuario 7	3	3	4	2	3	2	4	3	3	2	**62.5**
Usuario 8	5	1	5	1	4	2	5	1	4	1	**92.5**
Usuario 9	4	2	4	2	5	2	4	2	4	1	**80.0**
Usuario 10	5	2	5	1	5	1	5	1	5	1	**97.5**
Usuario 11	4	2	4	1	4	1	4	2	4	1	**82.5**
Usuario 12	5	1	4	2	4	2	5	1	4	2	**85.0**
PROMEDIO											**84.58**

En términos de desempeño general, el sistema alcanzó un Score Promedio SUS de **84.58**, una cifra que posiciona al agente virtual dentro de la categoría de usabilidad "Excelente". Este resultado se ve respaldado por la experiencia de la mayoría de los participantes, ya que 10 de los 12 usuarios consultados resaltaron tanto la facilidad de uso como la rapidez con la que se logra el aprendizaje de la herramienta.

Asimismo, la robustez del sistema quedó en evidencia al reportarse una baja necesidad de soporte técnico y una mínima presencia de inconsistencias durante su operación. No obstante, el análisis también permitió identificar áreas de oportunidad, específicamente a partir de un caso particular (*User* 7) que registró un puntaje de **62.5**. Este hallazgo señala la existencia de posibles puntos "engorrosos" en ciertos flujos de trabajo, los cuales serán sometidos a una revisión detallada para optimizar la experiencia del usuario.

5.9 Conclusión del Diseño Evaluado

El análisis exhaustivo y la convergencia de las métricas obtenidas fundamentan la recomendación técnica y operativa de implementar el agente virtual soportado en el modelo Claude Haiku 4.5, estructurado bajo la arquitectura de predominancia generativa expuesta en el Escenario 2 (E:0.3 / LLM:0.7). Esta resolución se sustenta en la priorización del rendimiento operativo sobre los costos de procesamiento en la nube, basándose en los siguientes hallazgos para cada modelo evaluado:

Desempeño del Agente Basado en Claude Haiku (Modelo Recomendado): Con la evaluación ejecutada sobre la muestra balanceada, esta variante técnica reportó la mejor puntuación de rendimiento general del estudio, logrando un *Accuracy* global del 64.0%. Su mayor virtud analítica se evidenció al detectar con éxito el 65.8% de las solicitudes ajenas a la infraestructura de red (reportes NO_ENERGETICO), contribuyendo de forma directa y tangible a mitigar el despilfarro de recursos humanos y logísticos que originan las visitas fallidas. Es fundamental destacar que, aunque el sistema aún se encuentra por debajo del hito institucional del 70% de asertividad global ,principalmente por la fricción analítica con la tipología de Podas, su despliegue es estratégicamente seguro y viable. Al retener correctamente el 93.6% de los reportes por fallas eléctricas reales, garantiza que las brigadas de Celsia puedan centrarse con celeridad en averías legítimas, eliminando el riesgo de ignorar cortes masivos o situaciones de peligro inminente y protegiendo así los indicadores de calidad del servicio.

Desempeño del Agente Basado en GPT-4o-mini (Alternativa Evaluada): En contraparte, la configuración más equilibrada para la arquitectura basada en OpenAI se situó en el Escenario 3 (Pesos Iguales). Bajo esta estructura, GPT-4o-mini exhibió un comportamiento algorítmico marcadamente conservador: ante la ambigüedad lingüística de los usuarios, el modelo optó sistemáticamente por clasificar el caso como una falla eléctrica legítima. Esta postura operativa resultó ser inherentemente segura, garantizando una retención sobresaliente de incidentes verdaderos (96.6%) y minimizando radicalmente el riesgo de desatender emergencias reales (apenas un 3.4% de falsos negativos). No obstante, este exceso de precaución penalizó de forma directa el objetivo central de la iniciativa. Al lograr interceptar únicamente el 53.6% de las visitas fallidas, su filtro semántico permitió que una alta proporción del ruido operativo (como suspensiones por mora o fallas internas del predio) atravesara hacia el sistema ADMS, amenazando con detonar nuevos despachos innecesarios de cuadrillas.

Costo Operativo vs. Valor de Negocio: A nivel transaccional, GPT-4o-mini demostró ser significativamente más económico, siendo 7.2 veces más barato por *token* que Claude Haiku 4.5. Sin embargo, a escala de producción (proyectando 86.000 reportes anuales), la diferencia absoluta es de apenas $85.10 USD al año. Dado que ambas tecnologías son extremadamente asequibles para el volumen operativo de Celsia (costando menos de $100 USD anuales), la brecha de precio en la API resulta insignificante frente al valor del negocio: el costo integral de realizar un solo despacho de brigada innecesario oscila entre $150 y $300 USD. Por lo tanto, la decisión

arquitectónica no debe fundamentarse en el costo de la infraestructura de IA, sino en el rendimiento operativo. Con una ventaja demostrada de 12.2 puntos porcentuales en la detección de visitas fallidas, se ratifica la selección de Claude Haiku 4.5, asumiendo su costo marginal a cambio de un retorno de inversión sustancialmente mayor para la operación.

6. CONCLUSIONES Y TRABAJOS FUTUROS

6.1 CONCLUSIONES

El desarrollo y validación del agente virtual inteligente para Celsia Colombia S.A. E.S.P. demostró empíricamente cómo la ciencia de datos y la inteligencia artificial generativa pueden transformar procesos operativos críticos en el sector eléctrico. De acuerdo con los objetivos planteados al inicio de esta investigación, se exponen las siguientes conclusiones:

6.1.1 Cumplimiento del Objetivo 1 - Diseño del flujo conversacional centrado en el cliente:

Se logró diseñar con éxito un flujo conversacional "*TO-BE*" apoyado en la metodología *Design Thinking.* Durante las fases de ideación y empatía, el equipo multidisciplinario aprendió que la alta tasa de error provenía de la falta de un diagnóstico guiado en el momento cero del reporte. Al implementar encuestas y validaciones directas con los operadores, se estructuró un flujo que exige al usuario detallar el evento (ej. identificar si los vecinos tienen luz o si hay chispas en el poste). Esto demostró que la recopilación activa de contexto semántico es el pilar para mitigar ambigüedades, logrando estructurar la información base para que el sistema reduzca las imprecisiones antes de la creación del tiquete en el ADMS.

6.1.2 Cumplimiento del Objetivo 2 - Construcción del sistema de clasificación y su rendimiento:

Se implementó con éxito un sistema de clasificación inteligente basado en una arquitectura híbrida y multi-agente, orquestada mediante `LangGraph` y `CrewAI`. La solución integra eficazmente reglas de negocio determinísticas con capacidades generativas de LLMs y búsqueda vectorial. En términos de rendimiento, la configuración con una predominancia del LLM (70%) resultó ser la más eficiente: alcanzó un tiempo de respuesta de 4.8 segundos por reporte, un *Accuracy* global del 64.0% y un F1-Macro de 0.5953. Estos resultados validan la superioridad del razonamiento de los LLMs frente a la clasificación semántica lineal tradicional.

Respecto al motor cognitivo, la evaluación comparativa demuestra que la elección del modelo impacta directamente en la rentabilidad operativa. Si bien modelos de bajo costo como GPT-4o-mini ofrecen un comportamiento conservador útil para reducir falsos negativos, la capacidad semántica de Claude Haiku 4.5 superó las expectativas (+12.2% en detección de visitas fallidas). Se concluye que, en procesos de triaje automatizado donde el costo de un error logístico (despacho innecesario de brigadas) supera el costo de procesamiento, maximizar la precisión justifica plenamente la inversión en un modelo con mayor costo por token.

6.1.3 Cumplimiento del Objetivo 3 - Validación del agente y reducción de visitas fallidas:

Se validó rigurosamente el agente virtual sobre una muestra histórica mixta y balanceada de 1.000 casos reales, confirmando su alto valor operativo. El sistema fue capaz de detectar de manera autónoma el 65.8% de los reportes correspondientes a "ruido" operativo (clase NO_ENERGETICO), como trámites de facturación o servicios ajenos a la compañía. Este nivel de detección certifica una reducción directa y tangible en la propensión de visitas técnicas fallidas. Además, la validación comprobó la seguridad del diseño propuesto, puesto que logró una tasa de retención del 93.6% sobre las fallas eléctricas genuinas (limitando los falsos negativos a tan solo 6.4%), garantizando así que la eficiencia operativa no comprometa la calidad y continuidad del servicio eléctrico.

6.2 TRABAJOS FUTUROS

A pesar de los sólidos resultados obtenidos, el análisis de las métricas de desempeño y la naturaleza dinámica del sector eléctrico plantean oportunidades claras de evolución para el sistema inteligente. De acuerdo con las necesidades técnicas y operativas del negocio, se recomiendan las siguientes líneas de trabajo a futuro:

6.2.1 Integración Real y Despliegue en Producción con LuzIA:

El paso más inmediato consiste en migrar el prototipo funcional actual hacia el entorno productivo de LuzIA en WhatsApp y los portales web de Celsia. Esta integración requerirá orquestar las APIs del agente propuesto con la infraestructura tecnológica que actualmente maneja el 93% de las interacciones automatizadas de la compañía, asegurando alta disponibilidad y concurrencia.

6.2.2 Mantenimiento del Modelo ante el Concept Drift:

El lenguaje natural utilizado por los clientes para reportar daños muta constantemente, introduciendo nuevos modismos, acrónimos o formas de referirse a infraestructuras recientes (como medidores AMI). Para evitar la degradación del *Accuracy* a lo largo del tiempo, se debe diseñar una estrategia de *Machine Learning Operations* (MLOps) que monitoree activamente el *concept drift.* Esto implica reentrenar periódicamente los *embeddings* y ajustar los *prompts* del LLM con muestras de conversaciones recientes.

6.2.3 Extensión a Otros Departamentos y Zonas de Operación de Celsia:

Dado que este proyecto validó los datos operativos correspondientes a Valle del Cauca y Tolima, se plantea como trabajo futuro escalar el modelo hacia otras regiones donde la compañía tenga o adquiera participación en distribución eléctrica. Esto exigirá parametrizar el sistema para

adaptarse a nuevas topologías de red, diferentes procesos de despacho y las particularidades léxicas de nuevas poblaciones.

6.2.4 Integración de Visión Computacional (Modelos Multimodales):

El principal desafío analítico del clasificador radica en la tipología de "Podas", la cual registró una baja tasa de detección (21.5%). Para dilucidar la ambigüedad entre "ramas rozando líneas" y "cortes severos causados por vegetación", se propone integrar capacidades multimodales que permitan analizar fotografías enviadas por los usuarios, combinando inferencia visual con texto.

6.2.5 Calibración Dinámica del Modelo:

A partir de los resultados comparativos entre el conservadurismo de modelos como GPT-4o-mini y la agresividad de detección de Claude Haiku, se propone como trabajo futuro la implementación de un umbral de confianza dinámico para el LLM Juez. Este mecanismo permitiría a Celsia ajustar el comportamiento del agente según la estacionalidad climática o picos de contingencia en la red eléctrica (ej., adoptar una postura conservadora durante tormentas severas para evitar filtrar falsos positivos, y retomar la configuración de alta intercepción de visitas fallidas durante la operación normal).

7. REFERENCIAS BIBLIOGRÁFICAS

[1]. IBM, “La IA en la experiencia del cliente (CX),” IBM Think, 2024. [En línea]. Disponible: https://www.ibm.com/es-es/think/topics/ai-customer-experience

[2]. Odigo, “3 formas de mejorar la experiencia de cliente con inteligencia artificial,” 2024. [En línea]. Disponible: https://www.odigo.com/es-es/blog-y-recursos/blog/3-formas-de-mejorar-la-experiencia-de-cliente-con-inteligencia-artificial

[3]. E. E. Sanabria y W. J. Frade, “Aportes de la inteligencia artificial al área de servicio al cliente,” Repositorio CUN, 2023. [En línea]. Disponible: https://repositorio.cun.edu.co/bitstream/handle/cun/6131/SanabriaWilson-2023-InteligenciaArtificialServicioalCliente.pdf

[4]. Red Hat, “Deep learning: qué es, usos, ventajas, su relación con machine learning,” 2023. [En línea]. Disponible: https://www.redhat.com/es/topics/ai/what-is-deep-learning

[5]. Aunoa, “El uso del machine learning en la atención al cliente,” 2024. [En línea]. Disponible: https://aunoa.ai/blog/el-machine-learning-en-la-atencion-al-cliente

[6]. Hack a Boss, “Cuáles los modelos de NLP aplicados al castellano,” 2025. [En línea]. Disponible: https://www.hackaboss.com/blog/cuales-son-los-modelos-de-nlp-procesamiento-del-lenguaje-natural-aplicados-al-castellano

[7]. Valtx, “Metodologías de desarrollo de software: ¿Qué son y para qué sirven?,” [En línea]. Disponible: https://www.valtx.pe/blog/metodologias-para-el-desarrollo-de-software-que-son-y-para-que-sirven

[8]. M. Luzardo y M. Hernández, “Implementación de un prototipo agente virtual,” Universidad Católica de Santiago de Guayaquil, Ecuador, 2020. [En línea]. Disponible: http://repositorio.ucsg.edu.ec/bitstream/3317/15673/1/T-UCSG-PRE-ING-CIS-278.pdf

[9]. ToGrow Agencia, “Fases del análisis y desarrollo de software: paso a paso,” 2025. [En línea]. Disponible: https://togrowagencia.com/fases-del-analisis-y-desarrollo-de-software/

[10]. ToGrow Agencia, “8 Metodologías de Desarrollo de Software para Ser Más Eficientes,” 2025. [En línea]. Disponible: https://togrowagencia.com/metodologias-de-desarrollo-de-software/

[11]. Design Thinking en español, “Qué es Design Thinking,” 2024. [En línea]. Disponible: https://designthinking.es/que-es-design-thinking

[12]. ClickUp, “Pasos del proceso Design Thinking & Ejemplos 2025,” 2024. [En línea]. Disponible: https://clickup.com/es-ES/blog/108135/proceso-de-diseno

[13]. AWS, “¿Qué es un LLM (modelo de lenguaje de gran tamaño) ?,” 2023. [En línea]. Disponible: https://aws.amazon.com/es/what-is/large-language-model

[14]. DataCamp, “Cómo funcionan los transformadores: Una exploración detallada de la arquitectura,” 2024. [En línea]. Disponible: https://www.datacamp.com/es/tutorial/how-transformers-work

[15]. IIC, “Transformers en Procesamiento del Lenguaje Natural,” 2024. [En línea]. Disponible: https://www.iic.uam.es/innovacion/transformers-en-procesamiento-del-lenguaje-natural

[16]. DataCamp, “Las 5 mejores bases de datos vectoriales | Una lista con ejemplos,” 2024. [En línea]. Disponible: https://www.datacamp.com/es/blog/the-top-5-vector-databases

[17]. IBM, “¿Qué es una base de datos vectorial?,” 2024. [En línea]. Disponible: https://www.ibm.com/es-es/think/topics/vector-database

[18]. Datitos, “Preprocesamiento de texto para NLP (parte 3),” 2020. [En línea]. Disponible: https://matiasbattocchia.github.io/datitos/Preprocesamiento-de-texto-para-NLP-parte-3.html

[19]. Impacto TIC, “IA en Colombia: Innovación y casos de éxito reciente,” 2025. [En línea]. Disponible: https://impactotic.co/inteligencia-artificial/ia-en-colombia-innovacion-y-casos-de-exito-reciente

[20]. OpenSistemas, “4 empresas innovadoras de inteligencia artificial en Colombia,” 2025. [En línea]. Disponible: https://opensistemas.com/inteligencia-artificial-en-colombia/

[21] UPME, "Informe de Redes Inteligentes y Gestión de Servicios en Colombia," 2024. [En línea]. Disponible en: https://www.upme.gov.co/siee/redes-inteligentes/

[22]. Santander Open Academy, “Metodologías de desarrollo de software: ¿qué son?,” 2025. [En línea]. Disponible: https://www.santanderopenacademy.com/es/blog/metodologias-desarrollo-software.html

[23]. J. Pavón, “Metodologías para el desarrollo de sistemas multi-agente,” Redalyc, [En línea]. Disponible: https://www.redalyc.org/pdf/925/92501805.pdf

[24] M. G. Soto, “Inteligencia Artificial: historia, construcción, modelos y lenguajes,” 2024.

[25] Telefónica Tech, “Una Breve Historia del Machine Learning,” 2024.

[26] ICX, “Design Thinking Y Diseño de experiencia del cliente,” 2022.

[27] Hayas Marketing, “La Inteligencia Artificial (IA) y su aplicación en Marketing,” 2025.

[28] BioAITeam, “Modelo de clasificación de incidencias mediante ML y NLP,” GitHub, 2024.

[29] Beetrack, “4 Ejemplos de Design Thinking de éxito,” 2024.

[30] InterSystems, “¿Qué son las bases de datos vectoriales y cómo funcionan?,” 2024.

[31] J. I. Baciero Fernández, “Sistema multiagente para el manejo de incidentes de seguridad,” UPM, 2021.

[32] Revista de Ciencias y Artes, “Inteligencia artificial (IA) y experiencia del cliente desde el año 2016,” 2024.

[33] J. I. Baciero Fernández, “Redes neuronales y NLP,” UPM, 2021.

[34] Maxim AI, "Best AI Agent Frameworks 2025: LangGraph, CrewAI, OpenAI,LlamaIndex, AutoGen," 2025. [En línea]. Disponible en: https://www.getmaxim.ai/articles/top-5-ai-agent-frameworks-in-2025

[35] T. H. Trần, "The AI Agent Framework Landscape in 2025: What Changed and What Matters," Medium, nov. 2025. [En línea]. Disponible en: https://medium.com/@hieutrantrung.it

[36] LangChain Inc., "LangGraph: Agent Orchestration Framework for Reliable AI Agents," 2024–2025. [En línea]. Disponible en: https://www.langchain.com/langgraph

[37] Latenode, "LangGraph Multi-Agent Orchestration: Complete Framework Guide + Architecture Analysis 2025," feb. 2026. [En línea]. Disponible en: https://latenode.com/blog/langgraph-multi-agent-orchestration

[38] Neurlcreators, "LangGraph 2025 Review: State-Machine Agents for Production AI," jul. 2025. [En línea]. Disponible en: https://neurlcreators.substack.com/p/langgraph-agent-state-machine-review

[39] K. Lin, "LangGraph: A Framework for Building Stateful Multi-Agent LLM Applications," Medium, mar. 2025. [En línea]. Disponible en: https://medium.com/@ken_lin/langgraph-a-framework-for-building-stateful

[40] Latenode, "LangGraph AI Framework 2025: Complete Architecture Guide + Multi-Agent Orchestration Analysis," feb. 2026. [En línea]. Disponible en: https://latenode.com/blog/langgraph-ai-framework-2025

[41] LangChain Inc., "LangGraph — GitHub Repository," 2024–2025. [En línea]. Disponible en: https://github.com/langchain-ai/langgraph

[42] CrewAI Inc., "Agents — CrewAI Documentation," 2025–2026. [En línea]. Disponible en: https://docs.crewai.com/en/concepts/agents

[43] Latenode, "CrewAI Framework 2025: Complete Review of the Open Source Multi-Agent AI Platform," feb. 2026. [En línea]. Disponible en: https://latenode.com/blog/crewai-framework

[44] Sider AI, "CrewAI Review 2025: Is This Multi-Agent Framework Worth Your Build?," 2025. [En línea]. Disponible en: https://sider.ai/blog/ai-tools/crewai-review-2025

[45] Codecademy, "Building Multi-Agent Applications with CrewAI," 2025. [En línea]. Disponible en: https://www.codecademy.com/article/multi-agent-application-with-crewai

[46] EmergentMind, "CrewAI Framework: Secure Multi-Agent Orchestration," 2025. [En línea]. Disponible en: https://www.emergentmind.com/topics/crewai

[47] CrewAI Inc., "The Open Source Multi-Agent Orchestration Framework," 2025–2026. [En línea]. Disponible en: https://crewai.com/open-source

[48] CrewAI Inc., "The Leading Multi-Agent Platform," 2025–2026. [En línea]. Disponible en: https://crewai.com

[49] EPM — Empresas Públicas de Medellín E.S.P., "Ema, la nueva asesora virtual de EPM," Sala de Prensa EPM, jun. 2019. [En línea]. Disponible en: https://www.epm.com.co/content/epm/institucional/sala-de-prensa/noticias-y-novedades/ema--la-nueva-asesora-virtual-de-epm.html

[50] EPM — Empresas Públicas de Medellín E.S.P., "Ema, el contacto digital de EPM, cumple cinco años, con más de 5 millones de interacciones con la comunidad," Sala de Prensa EPM, may. 2024. [En línea]. Disponible en: https://www.epm.com.co/content/epm/institucional/sala-de-prensa/noticias-y-novedades/ema-el-contacto-digital-de-epm-cumple-cinco-anos.html

[51] Minuto30, "EMA, la asistente virtual que transforma la atención al cliente en EPM," jun. 2025. [En línea]. Disponible en: https://www.minuto30.com/ema-la-asistente-virtual-que-transforma-la-atencion-al-cliente-en-epm/1664513/

[52] Congreso de la República de Colombia, "Ley 1581 de 2012, por la cual se dictan disposiciones generales para la protección de datos personales," Diario Oficial No. 48.587, oct. 2012. [En línea]. Disponible en: https://www.funcionpublica.gov.co/eva/gestornormativo/norma.php?i=49981

[53] EPM – Empresas Públicas de Medellín, "EMA: Asistente Virtual de EPM," EPM Sala de Prensa, jun. 2019. [En línea]. Disponible: https://www.epm.com.co

[54] EPM – Empresas Públicas de Medellín, "EMA WhatsApp supera 5.200.000 interacciones," EPM Sala de Prensa, mayo 2024. [En línea]. Disponible: https://www.epm.com.co

[55] EPM – Empresas Públicas de Medellín, "26% de interacciones digitales canalizadas a través de EMA," EPM Informe de Gestión Digital, 2024–2025. [En línea]. Disponible: https://www.epm.com.co

[56] Microsoft Research, "AutoGen: Enabling Next-Gen LLM Applications via Multi-Agent Conversation," 2024. [En línea]. Disponible en: https://microsoft.github.io/autogen/

[57] Microsoft Learn, "What is Semantic Kernel? Orchestrating AI with Skills and Planners," 2025. [En línea]. Disponible en: https://learn.microsoft.com/en-us/semantic-kernel/

[58] Deepset AI, "Haystack: The Open Source Framework for Building NLP Applications," 2025. [En línea]. Disponible en: https://haystack.deepset.ai/

[59] LlamaIndex, "Data Agents and Knowledge Orchestration Documentation," 2025. [En línea]. Disponible en: https://docs.llamaindex.ai/

[60] J. Martinez, "Comparative Analysis of Agentic Frameworks: From Logic to Data-Centric AI," AI Journal Tech, 2025.

[61] S. Gupta, "Architecting Multi-Agent Systems: A Deep Dive into AutoGen and Semantic Kernel," TechReview, 2026.

[62] Grupo Bancolombia, "Informe de Gestión y Sostenibilidad 2023," 2024. [En línea]. Disponible en: https://www.grupobancolombia.com/sostenibilidad/informes-gestion

[63] T. Brown, "Design Thinking," Harvard Business Review, vol. 86, no. 6, pp. 84-92, 2008. [En línea]. Disponible en: https://hbr.org/2008/06/design-thinking

[64] DNP, "CONPES 4144: Política Nacional de Inteligencia Artificial," Departamento Nacional de Planeación, 2024. [En línea]. Disponible en: https://www.dnp.gov.co/conpes

[65] ACL Anthology, "NLP Research in Latin America: Colombia Context," 2024. [En línea]. Disponible en: https://aclanthology.org

[66] A. M. Turing, "Computing Machinery and Intelligence," Mind, vol. 59, no. 236, pp. 433–460, oct. 1950.

[67] IEEE, "Automated Classification of Distribution System Faults using Natural Language Processing and Deep Learning," *IEEE Transactions on Power Systems*, 2024.

[68] Elsevier, "A text-mining-based framework for electric power distribution incident analysis and categorization," *Electric Power Systems Research*, 2025.

CONSENTIMIENTO INFORMADO PARA PARTICIPACIÓN EN SESIONES DE LEVANTAMIENTO DE INFORMACIÓN

Proyecto: Sistema de agentes virtuales para la clasificación automática de reportes de incidentes en Celsia Colombia S.A. E.S.P.

Institución: Pontificia Universidad Javeriana Cali — Maestría en Ciencia de Datos

Investigadores: María Fernanda Hernández Porras y Jesús David Sandoval Poso

Estimado/a participante:

Lo invitamos a participar en las sesiones de levantamiento de información del proyecto de investigación mencionado. Antes de confirmar su participación, le solicitamos leer detenidamente la siguiente información.

Propósito

Las sesiones tienen como objetivo recopilar información sobre el funcionamiento del ecosistema LuzIA, los flujos de clasificación de incidentes y las brechas operativas identificadas, con el fin de diseñar una solución basada en agentes virtuales inteligentes que mejore la precisión de clasificación y reduzca las visitas técnicas fallidas.

Participación

Su participación consiste en asistir a dos sesiones de aproximadamente una hora cada una, realizadas de manera virtual mediante Microsoft Teams. En cada sesión se realizarán preguntas sobre su rol, experiencia y conocimiento del sistema LuzIA, así como la revisión de casos operativos concretos.

Grabación y registro

Con su autorización, las sesiones serán grabadas para facilitar el análisis posterior. Las grabaciones serán de uso exclusivo del equipo investigador, no serán compartidas con terceros y serán eliminadas una vez concluido el proyecto.

Confidencialidad

La información recopilada será tratada de forma confidencial. Los datos operativos de Celsia mencionados durante las sesiones estarán disponibles para el evaluador académico únicamente bajo solicitud formal y con las condiciones de confidencialidad acordadas con la empresa.

Voluntariedad

Su participación es completamente voluntaria. Puede retirarse en cualquier momento sin que esto genere ninguna consecuencia para usted o para su relación con la empresa.

Contacto

Ante cualquier inquietud puede comunicarse con los investigadores a través de: [correo institucional de los autores].

Declaración de consentimiento

Declaro haber leído y comprendido la información anterior. Acepto participar voluntariamente en las sesiones descritas y autorizo la grabación de las mismas para los fines académicos indicados.

Nombre completo: Luis Alberto Carvajal Ramirez

Cargo: Análisista Experiencia al cliente

Empresa: Celsia Colombia S.A.

Firma: _____

Fecha: 25-05-2026

Fuente: Elaboración propia (2026).

Anexo A

CONSENTIMIENTO INFORMADO PARA PARTICIPACIÓN EN SESIONES DE LEVANTAMIENTO DE INFORMACIÓN

Proyecto: Sistema de agentes virtuales para la clasificación automática de reportes de incidentes en Celsia Colombia S.A. E.S.P.

Institución: Pontificia Universidad Javeriana Cali — Maestría en Ciencia de Datos

Autores: Jesús David Sandoval, María Fernanda Hernández

Estimado/a participante:

Lo invitamos a participar en las sesiones de levantamiento de información del proyecto de investigación mencionado. Antes de confirmar su participación, le solicitamos leer detenidamente la siguiente información.

Propósito

Las sesiones tienen como objetivo recopilar información sobre el funcionamiento del ecosistema LuzIA, los flujos de clasificación de incidentes y las brechas operativas identificadas, con el fin de diseñar una solución basada en agentes virtuales inteligentes que mejore la precisión de clasificación y reduzca las visitas técnicas fallidas.

Participación

Su participación consiste en asistir a dos sesiones de aproximadamente una hora cada una, realizadas de manera virtual mediante Microsoft Teams. En cada sesión se realizarán preguntas sobre su rol, experiencia y conocimiento del sistema LuzIA, así como la revisión de casos operativos concretos.

Grabación y registro

Con su autorización, las sesiones serán grabadas para facilitar el análisis posterior. Las grabaciones serán de uso exclusivo del equipo investigador, no serán compartidas con terceros y serán eliminadas una vez concluido el proyecto.

Confidencialidad

La información recopilada será tratada de forma confidencial. Los datos operativos de Celsia mencionados durante las sesiones estarán disponibles para el evaluador académico únicamente bajo solicitud formal y con las condiciones de confidencialidad acordadas con la empresa.

Voluntariedad

Su participación es completamente voluntaria. Puede retirarse en cualquier momento sin que esto genere ninguna consecuencia para usted o para su relación con la empresa.

Contacto

Ante cualquier inquietud puede comunicarse con los investigadores a través de: [correo institucional de los autores].

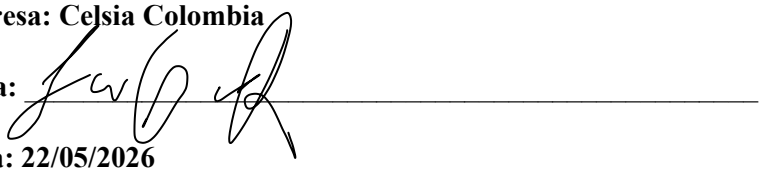
Declaración de consentimiento

Declaro haber leído y comprendido la información anterior. Acepto participar voluntariamente en las sesiones descritas y autorizo la grabación de estas para los fines académicos indicados.

Nombre completo: Sebastian Gómez Roldán

Cargo: Líder DevOps Comercial

Empresa: Celsia Colombia

Firma: 

Fecha: 22/05/2026

Fuente: Elaboración propia (2026).

CONSENTIMIENTO INFORMADO PARA PARTICIPACIÓN EN SESIONES DE LEVANTAMIENTO DE INFORMACIÓN

Proyecto: Sistema de agentes virtuales para la clasificación automática de reportes de incidentes en Celsia Colombia S.A. E.S.P.

Institución: Pontificia Universidad Javeriana Cali — Maestría en Ciencia de Datos

Investigadores: María Fernanda Hernández Porras y Jesús David Sandoval Poso

Estimado/a participante:

Lo invitamos a participar en las sesiones de levantamiento de información del proyecto de investigación mencionado. Antes de confirmar su participación, le solicitamos leer detenidamente la siguiente información.

Propósito

Las sesiones tienen como objetivo recopilar información sobre el funcionamiento del ecosistema LuzIA, los flujos de clasificación de incidentes y las brechas operativas identificadas, con el fin de diseñar una solución basada en agentes virtuales inteligentes que mejore la precisión de clasificación y reduzca las visitas técnicas fallidas.

Participación

Su participación consiste en asistir a dos sesiones de aproximadamente una hora cada una, realizadas de manera virtual mediante Microsoft Teams. En cada sesión se realizarán preguntas sobre su rol, experiencia y conocimiento del sistema LuzIA, así como la revisión de casos operativos concretos.

Grabación y registro

Con su autorización, las sesiones serán grabadas para facilitar el análisis posterior. Las grabaciones serán de uso exclusivo del equipo investigador, no serán compartidas con terceros y serán eliminadas una vez concluido el proyecto.

Confidencialidad

La información recopilada será tratada de forma confidencial. Los datos operativos de Celsia mencionados durante las sesiones estarán disponibles para el evaluador académico únicamente bajo solicitud formal y con las condiciones de confidencialidad acordadas con la empresa.

Voluntariedad

Su participación es completamente voluntaria. Puede retirarse en cualquier momento sin que esto genere ninguna consecuencia para usted o para su relación con la empresa.

Contacto

Ante cualquier inquietud puede comunicarse con los investigadores a través de: [correo institucional de los autores].

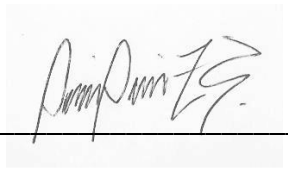
Declaración de consentimiento

Declaro haber leído y comprendido la información anterior. Acepto participar voluntariamente en las sesiones descritas y autorizo la grabación de las mismas para los fines académicos indicados.

Nombre completo: Darío Díaz Sánchez

Cargo: Análisis Tecnologías Operación TyD

Empresa: Celsia Colombia S.A.

Firma: 

Fecha: 11-05-2026

Fuente: Elaboración propia (2026).

Anexo 4. Guía de entrevista semiestructurada.

GUÍA DE ENTREVISTA SEMIESTRUCTURADA FASE DE EMPATÍA — DESIGN THINKING

Proyecto: Sistema de agentes virtuales para la clasificación automática de reportes de incidentes en Celsia Colombia S.A. E.S.P.

Institución: Pontificia Universidad Javeriana Cali — Maestría en Ciencia de Datos

Tipo de entrevista: Semiestructurada · Modalidad: Virtual (Microsoft Teams) · Duración estimada: 1 hora por sesión

Instrucciones para el entrevistador

Las preguntas que conforman esta guía son orientadoras. El entrevistador puede adaptar el orden y la profundidad de cada bloque según el perfil del participante. Se recomienda iniciar con preguntas de contexto antes de abordar las de problemas y brechas. Ante respuestas relevantes no previstas, se alienta al entrevistador a profundizar con preguntas de seguimiento.

Bloque 1 — Contexto y rol

1. ¿Cuál es su rol dentro del proceso de atención de reportes de incidentes en Celsia?

2. ¿Desde cuándo interactúa con el ecosistema LuzIA y qué etapas de su evolución ha vivido directamente?

3. ¿Cómo describiría el flujo actual desde que un cliente reporta un incidente hasta que se crea el evento en el sistema de gestión?

Bloque 2 — Funcionamiento del sistema actual

1. ¿Cuáles son los criterios que actualmente usa el sistema para clasificar un reporte como daño de energía?

2. ¿Cómo funciona la integración entre el canal de WhatsApp y los sistemas internos de gestión de incidentes? ¿Qué datos viajan entre esos sistemas?

3. ¿Qué ocurre cuando el sistema no logra clasificar una conversación? ¿Qué ve el cliente y qué sucede internamente?

Bloque 3 — Problemas y brechas identificadas

1. ¿Cuáles son las tipologías de reportes que con mayor frecuencia generan clasificaciones incorrectas?

2. ¿Qué información aporta el cliente en el texto libre que el sistema actualmente ignora o no aprovecha?

3. ¿Cuál es el impacto concreto de una visita fallida en la operación del Centro de Control?

4. ¿Qué errores recurrentes se presentan en los datos del cliente al ser registrados en el sistema de gestión de incidentes?

Bloque 4 — Experiencia del cliente y percepción del servicio

1. ¿Cómo percibe el cliente el flujo conversacional actual? ¿Qué quejas recurrentes se reciben?

2. ¿Qué preguntas diagnósticas considera que deberían incluirse en el flujo para mejorar la clasificación?

3. ¿En qué municipios o zonas se concentra la mayor proporción de visitas fallidas y por qué?

Bloque 5 — Visión de la solución

1. Si pudiera rediseñar el flujo conversacional, ¿qué cambiaría primero y por qué?

2. ¿Qué restricciones técnicas u operativas considera más importantes al diseñar una solución de clasificación automática?

3. ¿Qué métricas usaría para evaluar si la solución propuesta realmente mejoró el proceso?

Fuente: Elaboración propia a partir de las sesiones de levantamiento con equipos de Celsia (2026).