Escuela Superior Politécnica del Litoral

Facultad de Ingeniería en Electricidad y Computación

Sistema de gestión veterinaria automatizado con inteligencia conversacional y visualización de datos en tiempo real

Código de proyecto

INGE-2985

Proyecto Integrador

Previo la obtención del Título de:

Ingeniero en Telemática

Presentado por:
Francisco José Terán Cobo

Erick David Flores Campaña

GUAYAQUIL – ECUADOR

Año: 2025

Dedicatoria

Dedico este proyecto a mi familia, por su confianza, paciencia y respaldo incondicional en cada etapa de mi vida académica. A quienes creyeron en mí y me brindaron fuerzas para superar desafíos y cumplir este importante objetivo profesional.

Francisco Teran Cobo

Dedico este logro a mis padres, por su amor incondicional, apoyo incondicional, apoyo constante y sacrificios realizados a lo largo de mi formación. A mi familia, por ser mi motivación diaria y el pilar fundamental que me impulsó a alcanzar esta meta.

Erick Flores Campaña

Agradecimientos

Agradezco profundamente al M.Sc. Danny Torres por su valiosa guía, paciencia y apoyo constante durante el desarrollo de esta tesis. Asimismo, expreso mi gratitud a los profesores de ESPOL por sus enseñanzas y compromiso con mi formación profesional.

Francisco Teran Cobo

Mi sincero agradecimiento al M.Sc. Danny Torres por su orientación académica y acompañamiento en este proceso. También agradezco a los profesores de la ESPOL por compartir sus conocimientos y contribuir significativamente a mi crecimiento académico y profesional.

Erick Flores Campaña

Declaración Expresa

Nosotros Francisco José Teran Cobo y Erick David Flores Campaña acordamos y reconocemos que:

La titularidad de los derechos patrimoniales de autor (derechos de autor) del proyecto de graduación corresponderá al autor o autores, sin perjuicio de lo cual la ESPOL recibe en este acto una licencia gratuita de plazo indefinido para el uso no comercial y comercial de la obra con facultad de sublicenciar, incluyendo la autorización para su divulgación, así como para la creación y uso de obras derivadas. En el caso de usos comerciales se respetará el porcentaje de participación en beneficios que corresponda a favor del autor o autores.

La titularidad total y exclusiva sobre los derechos patrimoniales de patente de invención, modelo de utilidad, diseño industrial, secreto industrial, software o información no divulgada que corresponda o pueda corresponder respecto de cualquier investigación, desarrollo tecnológico o invención realizada por nosotros durante el desarrollo del proyecto de graduación, pertenecerán de forma total, exclusiva e indivisible a la ESPOL, sin perjuicio del porcentaje que nos corresponda de los beneficios económicos que la ESPOL reciba por la explotación de nuestra innovación, de ser el caso.

En los casos donde la Oficina de Transferencia de Resultados de Investigación (OTRI) de la ESPOL comunique a los autores que existe una innovación potencialmente patentable sobre los resultados del proyecto de graduación, no se realizará publicación o divulgación alguna, sin la autorización expresa y previa de la ESPOL.

Guayaquil, 12 de febrero del 2025.

Firmado electrónicamente por:
FRANCISCO JOSE TERAN COBO
Validar únicamente con FirmaEC

Francisco José Teran Cobo

Firmado electrónicamente por:
ERICK DAVID FLORES CAMPANA
Validar únicamente con FirmaEC

Erick David Flores Campaña

Evaluadores

Firmado electrónicamente por:
CHRISTOPHER JAVIER VACCARO CEDILLO
Validar únicamente con FirmaEC

M.Sc. Christopher Javier Vaccaro Cedillo

Profesor de Materia

Firmado electrónicamente por:
DANNY ALFREDO TORRES MORAN
Validar únicamente con FirmaEC

M.Sc. Danny Alfredo Torres Morán

Tutor de proyecto

Resumen

En las clínicas veterinarias de atención ambulatoria, la gestión manual de citas y consultas a través de mensajería instantánea genera retrasos, sobrecarga operativa y riesgo de errores en la programación. La falta de centralización de la información y de herramientas automatizadas limita la eficiencia del personal y afecta la experiencia del tutor de la mascota. Ante esta problemática, el presente proyecto propone el desarrollo de una plataforma integral de gestión y soporte al cliente basada en automatización e inteligencia conversacional. Se plantea como hipótesis que la integración de flujos de trabajo automatizados y modelos de lenguaje permite optimizar la administración de citas y reducir tareas repetitivas. El sistema fue diseñado bajo una arquitectura en capas e implementado mediante contenedores Docker. Se emplearon n8n para la orquestación de flujos, Evolution API para la integración con mensajería, MongoDB como base de datos transaccional, Qdrant para almacenamiento vectorial y modelos de OpenAI para clasificación y extracción de información. Asimismo, se incorporaron servicios en la nube para almacenamiento y despliegue. Los resultados mostraron un nivel de automatización superior al 80% en la gestión de citas y una reducción significativa de intervención manual. Se concluye que la automatización inteligente mejora la eficiencia operativa y fortalece la organización del servicio clínico.

Palabras Clave: automatización de procesos, inteligencia conversacional, arquitectura en capas, mensajería digital, eficiencia operativa.

Abstract

In outpatient veterinary clinics, manual management of appointments and inquiries through instant messaging generates delays, operational overload, and a higher risk of scheduling errors. The lack of centralized information and automated tools limits staff efficiency and negatively impacts the pet owner's experience. In response to this issue, this project proposes the development of an integrated management and customer support platform based on automation and conversational intelligence. The hypothesis states that integrating automated flujos de trabajos and language models improves appointment administration and reduces repetitive tasks. The system was designed using a layered architecture and implemented through Docker containers. n8n was used for workflow orchestration, Evolution API enabled messaging integration, MongoDB supported transactional data storage, Qdrant functioned as a vector database, and OpenAI models were applied for message classification and entity extraction. Cloud services were incorporated for storage and deployment. Results showed an automation level above 80% in appointment management and a significant reduction in manual intervention. It is concluded that intelligent automation enhances operational efficiency and strengthens clinical service organization.

Keywords: process automation, conversational intelligence, layered architecture, digital messaging, operational efficiency.

Índice general

Abreviaturas

API	Interfaz para comunicación entre sistemas.
ASR	Reconocimiento automático del habla.
BPM	Gestión de procesos de negocio.
DAG	Grafo dirigido acíclico.
E2E	Flujo completo de extremo a extremo.
FAQ	Preguntas frecuentes.
GPT	Modelo de lenguaje generativo basado en transformadores.
HTTP	Protocolo de transferencia web.
IA	Inteligencia Artificial.
IoT	Internet de las Cosas.
NLP	Procesamiento del lenguaje natural.
OCR	Reconocimiento óptico de caracteres.
QR	Código de respuesta rápida.
RAG	Generación aumentada por recuperación de información.
RPA	Automatización robótica de procesos.
SaaS	Software como servicio.
SGBD	Sistema de gestión de bases de datos.
SMS	Servicio de mensajería de texto.
TTS	Síntesis de texto a voz.
UI	Interfaz de usuario.
UX	Experiencia del usuario.
VPN	Red privada virtual.

Índice de figuras

Índice de tablas

Capítulo 1

1.1 Introducción

En las últimas décadas, la digitalización ha transformado diversos sectores; sin embargo, en la medicina veterinaria su adopción es aún limitada y desigual. En Bosnia y Herzegovina, por ejemplo, se reportó que solo el 10,2% de las organizaciones veterinarias cuentan con un sitio web, a pesar de que más del 70% de la población utiliza Internet, evidenciando una brecha tecnológica común también en Latinoamérica (Fejzic et al., 2024).

En Ecuador, esta problemática persiste: la mayoría de las clínicas carece de sistemas centralizados y mantiene registros en formatos físicos (Ochoa et al., 2020). Esta gestión manual deriva en desorden administrativo, pérdida de datos y retrasos en la atención, mientras que el uso de agendas físicas incrementa los conflictos de horarios y la sobrecarga laboral (Cachuela et al., 2025). Particularmente en Guayaquil, la dependencia de canales informales y registros manuales reduce drásticamente la eficiencia y la trazabilidad operativa.

Ante la falta de sistemas integrados para consolidar historiales y gestionar la comunicación, este proyecto propone desarrollar una plataforma que unifique la administración de citas, la organización clínica y la interacción con los propietarios. El objetivo es optimizar la eficiencia operativa, minimizar errores administrativos y mejorar sustancialmente la experiencia del usuario.

1.2 Descripción del Problema

En numerosas clínicas veterinarias, la gestión de citas y registros clínicos se realiza aún de forma manual o mediante herramientas no integradas. En Ecuador, estudios realizados en Guayaquil y Quito confirman que el uso de agendas físicas y formularios impresos genera pérdida de tiempo, dificultades en la recuperación de datos y baja trazabilidad (Loor García, 2019; Ochoa et al., 2020; Vela Velastegui, 2020).

Esta dependencia de métodos tradicionales satura al personal con tareas repetitivas —como la búsqueda de expedientes y la confirmación manual de citas—, lo que eleva el riesgo de errores operativos y compromete la calidad del servicio (Ochoa et al., 2020).

La problemática se agudiza ante la ausencia de un canal conversacional automatizado que esté integrado al sistema de gestión. Sin esta tecnología, la atención de consultas y el agendamiento dependen de llamadas telefónicas en horarios restringidos, provocando esperas y abandono de clientes potenciales que demandan atención omnicanal 24/7. Esta limitación no solo deteriora la experiencia del usuario, sino que reduce la competitividad frente a centros digitalizados.

Por tanto, el desafío radica en la incapacidad operativa para gestionar solicitudes masivas en tiempo real sin incrementar la carga laboral. Resulta imperativo integrar soluciones de inteligencia conversacional en mensajería instantánea que automaticen procesos recurrentes y permitan el escalamiento humano cuando la situación lo amerite.

De lo contrario, las clínicas permanecerán vulnerables a cuellos de botella y picos de demanda, aumentando el estrés del equipo y la probabilidad de errores en la atención (Balto, 2025; Molloy, 2025).

1.3 Justificación del problema

La implementación de plataformas tecnológicas en clínicas veterinarias optimiza la eficiencia operativa al automatizar tareas rutinarias y reducir errores administrativos (Fejzic et al., 2024; Kammrath Betancor et al., 2025). Específicamente, los sistemas web de gestión de citas centralizan la información clínica y disminuyen el ausentismo mediante recordatorios automáticos, lo que fortalece la experiencia del usuario (Abqari et al., 2022; Ochoa et al., 2020). En Ecuador, experiencias en Santa Elena y Durán confirman que la digitalización agiliza la organización interna, reduce la pérdida de datos y mejora la precisión de los historiales clínicos (Chávez Yagual, 2022; Gregorio & Angel, 2021; Cachuela et al., 2025).

Desde un enfoque ingenieril, este proyecto propone una arquitectura integrada que resuelve la atención al cliente y la operación clínica de forma coherente. La solución incluye un canal conversacional 24/7 capaz de gestionar solicitudes frecuentes —como agendamiento y reprogramación— y escalar a atención humana según sea necesario. Esto garantiza la continuidad del servicio sin incrementar la carga laboral del personal ni generar cuellos de botella por intervención manual.

Finalmente, la integración entre mensajería, agenda y registros elimina la duplicidad de datos y asegura la trazabilidad de las interacciones. Al fundamentar las respuestas de la IA en información interna, se garantiza la confiabilidad del servicio. Así, esta plataforma no solo moderniza la gestión veterinaria, sino que constituye una solución técnica escalable que eleva la competitividad en un entorno que demanda inmediatez.

1.4 Objetivo general

Desarrollar un sistema inteligente para la gestión de citas en una empresa de servicios veterinarios, incorporando un asistente conversacional para la interacción con clientes, la automatización de procesos internos y la visualización accesible de la información.

1.5 Objetivos específicos

- Integrar módulos de inteligencia artificial para agendar citas y enviar notificaciones automáticas, con el fin de optimizar la gestión operativa y la atención al cliente.
- Construir una base de conocimiento dinámica e interactiva que proporcione respuestas contextuales y personalizadas al cliente, integrando información actualizada de los sistemas internos, garantizando un nivel de precisión y fiabilidad en la información.
- Desarrollar un módulo de visualización física que presente en tiempo real la información relevante sobre las citas para la organización interna.

1.6 Alcance del Proyecto

El proyecto contempla el desarrollo e implementación de un sistema integral de gestión veterinaria que articula automatización de procesos, inteligencia conversacional y monitoreo de datos en tiempo real. El alcance técnico incluye:

- **Asistente Conversacional Inteligente:** Desarrollo de un agente basado en modelos de lenguaje de OpenAI, integrado con una base de datos vectorial (Qdrant) y MongoDB para la recuperación de información personalizada (RAG) y gestión de historiales.

- **Gestión Automatizada de Citas:** Sistema de agendamiento y reprogramación con envío de notificaciones automáticas mediante integraciones con la API de Google y Evolution API.
- **Módulo de Visualización Física:** Implementación de una interfaz de monitoreo con actualización de datos en tiempo real y latencia inferior a cinco segundos.
- **Infraestructura y Despliegue:** Orquestación de servicios mediante contenedores Docker y Portainer, utilizando Redis como gestor de colas de mensajes y n8n para la automatización de flujos de trabajo.
- **Seguridad y Disponibilidad:** Alojamiento en servidor dedicado con configuración de proxies y VPN para garantizar la integridad de los datos y la estabilidad de las comunicaciones.

1.7 Limitaciones del Proyecto

El desarrollo del sistema está sujeto a las siguientes restricciones técnicas y operativas:

- **Dependencia de la Base de Conocimiento:** La precisión del asistente conversacional está supeditada a la calidad y actualización de los datos registrados; información inconsistente o inexistente limitará la pertinencia de las respuestas.
- **Factores de Infraestructura:** El rendimiento y la sincronización en tiempo real dependen directamente de la estabilidad de la conexión a internet y de la capacidad de procesamiento del servidor contratado.
- **Servicios de Terceros:** La continuidad operativa de las integraciones (Evolution API, Google APIs y OpenAI) está condicionada a la disponibilidad, cambios en las políticas o actualizaciones de dichos proveedores externos.
- **Restricción del Módulo Físico:** El hardware de visualización está limitado al entorno de red local de la clínica, requiriendo una conexión estable para garantizar latencias menores a cinco segundos.

- **Ciberseguridad:** Si bien se implementan capas de seguridad mediante proxies y VPN, la operación no es inmune a vulnerabilidades emergentes o ataques externos sofisticados fuera del control del sistema.

1.8 Estado del arte

La evolución de los sistemas conversacionales es uno de los avances más relevantes de la IA contemporánea. Los primeros prototipos, como ELIZA, demostraron que era posible simular una conversación, pero sin comprensión semántica: su funcionamiento se basaba en coincidencias de patrones y transformaciones superficiales del texto (Weizenbaum, 1966).

Con la transición hacia redes neuronales y, posteriormente, arquitecturas Transformer, los asistentes actuales han mejorado de forma sustancial al interpretar intención, manejar contexto extendido y generar respuestas coherentes, incluso en dominios especializados.

Los asistentes modernos, apoyados en Modelos de lenguaje natural y aprendizaje profundo, superan ampliamente a los sistemas basados en reglas. Se ha reportado que pueden adaptar el tono, aprender de interacciones y mantener coherencia contextual, lo que los hace adecuados para escenarios donde la naturalidad y precisión son críticas, incluyendo ámbitos de salud y servicios (Cole, 2024).

Además, la adopción de soluciones conversacionales ha crecido por su disponibilidad 24/7, menor costo operativo y facilidad de interacción en lenguaje natural (Oracle, 2020).

En el sector veterinario, durante los últimos cinco años se observa una adopción creciente de estas tecnologías asociada a la profesionalización del servicio y a la mejora de la comunicación con los propietarios. Se ha evidenciado que la digitalización y automatización administrativa optimiza la atención y eleva la calidad del servicio (Dmitrievich & Alexandrovich, 2024). También se han documentado beneficios concretos de los recordatorios automatizados (SMS/correo), como la reducción de ausencias y un seguimiento más continuo de pacientes (Covetrus, 2025).

Un foco recurrente en la literatura es la gestión de citas y la comunicación administrativa, debido al alto volumen de consultas, reprogramaciones y preguntas frecuentes.

En América Latina, se reportan sistemas de agendamiento automatizado que reducen tiempos de espera y mejoran la disponibilidad del personal, aunque varios aún dependen de reglas fijas sin incorporar IA generativa ni mecanismos avanzados de contextualización (Cotrina Cabezas, 2022).

Otros análisis señalan que el agendamiento automático reduce carga administrativa y cancelaciones tardías, pero persiste como limitación la escasa integración con fuentes internas de conocimiento (Arifah Fasha et al., 2023; VetSoftwareHub, 2025).

Ante ello, ha aumentado el interés en Recuperación Aumentada por Generación (RAG), que combina modelos generativos con recuperación de información para fundamentar respuestas en documentos externos actualizados.

La evidencia reciente indica que RAG reduce alucinaciones y mejora la exactitud en dominios especializados, y que su rendimiento aumenta cuando incorpora guías, manuales o registros internos del dominio (Kulkarni et al., 2024; Li et al., 2025). En paralelo, la automatización mediante flujos de trabajo permite integrar chatbots con calendarios, sistemas de gestión y mensajería. En entornos veterinarios se asocia con centralización de datos, menor error humano y mayor consistencia en comunicación y seguimiento (Beyer, 2023; Vetstoria, 2024).

Otro eje emergente es la interacción multimodal (texto, voz e imágenes). Se ha señalado que la multimodalidad ayuda a resolver ambigüedades al combinar canales, lo cual es útil cuando los propietarios describen síntomas por voz o envían fotografías (Fernández, 2025).

Estudios aplicados muestran mejoras en detección temprana en contextos como dermatología veterinaria mediante análisis de imágenes, y avances en agentes capaces de interpretar consultas habladas y contenido visual para responder de forma contextualizada (Gadiraju et al., 2024; Huong et al., 2025).

Pese a estos avances, la literatura coincide en que no existe aún una solución integral que unifique en un mismo ecosistema: agendamiento automatizado, gestión inteligente de clientes, RAG, automatización de flujos y capacidades multimodales.

La mayoría de los trabajos aborda solo uno o dos componentes de forma aislada, lo que deja una brecha para propuestas que articulen estas líneas dentro de un flujo conversacional coherente y alineado con necesidades operativas reales.

En este marco, el presente proyecto plantea integrar modelos generativos, RAG, automatización de procesos, gestión contextualizada de citas y multimodalidad para superar limitaciones reportadas: dependencia de reglas, baja personalización, poca integración entre sistemas y dificultad para operar información clínica/administrativa en tiempo real.

1.9 Marco teórico

1.9.1 Fundamentos de inteligencia artificial

- **Inteligencia Artificial (IA):** La Inteligencia Artificial (IA) es una rama de la informática que se enfoca en la creación de sistemas capaces de realizar tareas que normalmente requieren inteligencia humana.
  Estas tareas incluyen el razonamiento, la percepción, el aprendizaje, la resolución de problemas y el procesamiento del lenguaje natural (Vetstoria, 2024). A lo largo de las décadas, la IA ha evolucionado desde algoritmos basados en reglas y lógica hasta sistemas modernos que aprenden automáticamente a partir de datos mediante técnicas como el aprendizaje automático (Machine Learning) y el aprendizaje profundo (Deep Learning).
  Esta capacidad ha permitido avances significativos en sectores como la salud, la educación, el transporte, la robótica y la atención al cliente, donde los sistemas pueden predecir, clasificar o incluso conversar de manera autónoma (Stryker & Kavlakoglu, s. f.).

- **Modelos de inteligencia artificial:** Un modelo de inteligencia artificial es una representación matemática entrenada para identificar patrones o inferencias a partir de datos. Los modelos de IA aprenden reglas a través de ejemplos, y se ajustan continuamente con nuevos datos o retroalimentación para mejorar su precisión. Entre los tipos principales destacan:

  - **Modelos supervisados**: entrenados con datos etiquetados (por ejemplo, clasificación de correos spam).

- **No supervisados:** detectan patrones sin etiquetas previas (por ejemplo, agrupamiento o clustering).
- **Modelos de lenguaje natural (NLP):** especializados en procesar y generar lenguaje humano, como los modelos BERT, T5 o GPT.
- **Modelos generativos:** crean contenido nuevo como texto, imágenes, audio o video. Ejemplo: GPT-4, DALL·E, Stable Diffusion (Brown et al., 2020).
- Actualmente, modelos multimodales como GPT-4 o Gemini son capaces de procesar texto, imagen y voz en una sola arquitectura, marcando un hito en la evolución de los sistemas inteligentes (OpenAI, 2024).

- **Chatbots e inteligencia artificial conversacional:** Un chatbot es un programa informático diseñado para simular conversaciones humanas de forma natural, ya sea mediante texto o voz (Oracle, 2020). En su nivel más básico, un chatbot procesa la entrada del usuario (texto escrito o habla) y produce respuestas automáticas que imitan el diálogo humano (Oracle, 2020).

  En esencia, la IA conversacional busca que la interacción hombre-máquina sea lo más similar posible a una conversación entre personas, ofreciendo respuestas coherentes, personalizadas y disponibles en cualquier momento (Cole, 2024).

  • Modelos de lenguaje GPT(Generative Pre-trained Transformer): Es una familia de modelos de lenguaje desarrollados por OpenAI, diseñados para generar texto coherente y contextualizado a partir de una entrada dada.

  Utiliza la arquitectura de *transformer*, introducida por (Vaswani et al., 2023), y se entrena previamente con grandes volúmenes de texto para luego ajustarse a tareas específicas (Brown et al., 2020).

  GPT-4, la versión más reciente, no solo comprende y genera texto, sino que puede analizar imágenes, escribir código y mantener diálogos prolongados con contexto.

  A diferencia de modelos tradicionales, GPT no solo responde, sino que puede redactar ensayos, traducir idiomas, responder preguntas técnicas y generar texto creativamente (OpenAI, 2024). Su capacidad para "razonar" se basa en patrones

estadísticos y contexto semántico, no en comprensión real del mundo, pero su desempeño ha superado a muchos modelos anteriores en tareas de comprensión lectora, escritura técnica y conversación.

1.9.2 Recuperación de conocimiento y búsqueda semántica

- **Recuperación aumentada por generación (RAG):** La Recuperación Aumentada por Generación (traducido al inglés: *Retrieval-Augmented Generation*) es una técnica que combina modelos de lenguaje natural con la búsqueda de información en una base de conocimiento externa. Su objetivo es mejorar la precisión y confiabilidad de las respuestas, complementando al modelo con datos específicos y actualizados del dominio (Merritt, 2025). A diferencia de los modelos tradicionales, cuyo conocimiento es estático tras el entrenamiento, RAG primero recupera documentos relevantes mediante búsqueda semántica y luego utiliza ese contenido como contexto para generar la respuesta (Nebot, 2025). Esta búsqueda se apoya en representaciones vectoriales, que permiten encontrar información relacionada por significado y no solo por coincidencia exacta de palabras, superando limitaciones de la búsqueda por palabras clave (Nebot, 2025). En síntesis, RAG conecta un modelo generativo con un sistema de recuperación de conocimientos para producir respuestas mejor fundamentadas, reducir alucinaciones y aumentar la calidad en tareas intensivas en información, especialmente en entornos empresariales que requieren contenido especializado (Lewis et al., 2020).

- **Sistemas de recuperación de conocimiento:** Un sistema de recuperación de conocimientos es un conjunto de tecnologías que permite almacenar, indexar y recuperar información relevante para una consulta, normalmente con el objetivo de responder preguntas o asistir en la toma de decisiones (Devlin et al., 2018). Estos sistemas se basan en tres componentes esenciales:

    - Una base de conocimiento: documentos, artículos, FAQs o registros organizados.

    - Un motor de búsqueda: que puede ser por coincidencia de texto o semántica (usando embeddings vectoriales).

- Un modelo de lenguaje o interfaz: que interpreta la consulta del usuario y presenta una respuesta.
- La integración de modelos generativos como GPT con estos sistemas da lugar a soluciones más sofisticadas como RAG, en las que el modelo no responde solo con su entrenamiento, sino que consulta fuentes externas antes de generar la salida (Lewis et al., 2020).
- Estos sistemas tienen múltiples aplicaciones en servicio al cliente, soporte técnico, educación y medicina.

- **Embeddings y bases de datos vectoriales:** La búsqueda semántica y RAG se apoyan principalmente en dos elementos: *embeddings* y *bases de datos vectoriales*.

  Los *Vectores de características* (traducido al inglés como *embeddings*) son representaciones vectoriales numéricas que capturan el significado de textos u otros datos. En lenguaje natural, permiten que conceptos similares queden "cerca" en un espacio vectorial (por ejemplo, "perro" más próximo a "gato" que a "automóvil"), lo que hace posible medir similitud por significado y no por coincidencia exacta de palabras (Nebot, 2025).

  Las bases de datos vectoriales almacenan e indexan estos vectores para realizar búsquedas de similitud de forma eficiente. A diferencia de bases relacionales, están optimizadas para encontrar rápidamente los embeddings más cercanos a una consulta, lo que habilita la búsqueda semántica a escala (Aquino, 2024).

  Tecnologías como Qdrant o Milvus permiten consultas en tiempo real incluso con grandes volúmenes de vectores (Qdrant, 2023).

  En este proyecto, una base vectorial como Qdrant puede guardar embeddings de contenidos relevantes de la clínica (FAQs, políticas, información de servicios, etc.) para recuperar en milisegundos los fragmentos más relacionados con la pregunta del cliente y apoyar respuestas más contextualizadas mediante RAG.

1.9.3 Automatización e integración de procesos

- **Automatización de flujos con herramientas no-code (n8n):** *n8n* es una plataforma de automatización de flujos de trabajo de código abierto (enfoque low-code/no-code) que permite orquestar integraciones entre distintos sistemas mediante una interfaz visual basada en nodos (Oriigin, 2025).

  En lugar de programar cada conexión desde cero, ofrece componentes predefinidos (por ejemplo, para APIs, calendarios o mensajería) que se enlazan para construir procesos con disparadores, reglas y acciones.

  Su principal valor en este proyecto es conectar el canal conversacional con los servicios internos y externos de la clínica para ejecutar tareas de forma consistente y automática.

  Al ser auto-hospedable, también permite mayor control sobre datos e infraestructura, lo cual es relevante en entornos que manejan información sensible (Oriigin, 2025).

- **Webhooks y triggers:** Un webhook es un mecanismo que permite que un sistema notifique automáticamente a otro cuando ocurre un evento específico, enviando una solicitud HTTP a una URL determinada (Twilio, 2023). Un trigger (disparador) es la condición o evento que activa el webhook. En el proyecto, los triggers pueden ser: "cliente envió mensaje por WhatsApp", "se subió un archivo" o "se creó una cita". Esos eventos disparan flujos en n8n, activando respuestas automáticas o almacenamiento en la base vectorial.

- **Sistemas de mensajería RabbitMQ**: RabbitMQ es un sistema de mensajería orientado a colas (message broker) que permite desacoplar componentes del sistema mediante el envío asíncrono de mensajes (Pivotal, 2023). Permite que un servicio publique un evento (por ejemplo, "documento recibido") y que otro servicio lo procese más tarde sin conexión directa. En el proyecto, se usa para distribuir eventos como mensajes recibidos, inicios de flujos o confirmaciones de cita, mejorando la escalabilidad y la tolerancia a fallos del sistema.

- **Hiperautomatización:** La hiper automatización es un enfoque estratégico que combina múltiples tecnologías de automatización como RPA (Robotic Process Automation), inteligencia artificial, aprendizaje automático y herramientas no-code para automatizar procesos empresariales de extremo a extremo (Definition of Hyperautomation, s. f.). A diferencia de la automatización

tradicional, que se limita a tareas repetitivas y estructuradas, la hiper automatización busca automatizar procesos complejos que requieren toma de decisiones, comprensión del lenguaje o adaptación dinámica (What Is Hyperautomation? | IBM, s. f.). Esto se logra mediante la integración de sistemas de IA conversacional, motores de flujos (como n8n o Zapier), análisis predictivo y bases de conocimiento que permiten que los sistemas no solo ejecuten tareas, sino que aprendan, se adapten y colaboren con humanos. Por ejemplo, en el contexto del presente proyecto, un flujo automatizado podría recibir una solicitud de cita desde WhatsApp, verificar la disponibilidad, acceder a la base de datos, enviar la confirmación al cliente y generar un resumen para el veterinario, todo sin intervención humana. La hiperautomatización representa, por tanto, la evolución natural de la transformación digital: pasar de automatizar tareas aisladas a orquestar inteligentemente procesos completos (Definition of Hyperautomation, s. f.).

1.9.4 Arquitectura y desarrollo del sistema

- **Backend:** El backend es la parte del sistema que gestiona la lógica, procesamiento de datos, autenticación, y conexiones con la base de datos o servicios externos. Opera en el servidor y no es visible directamente para el usuario final (Jovanovic et al., 2020). En este proyecto, el backend recibe las solicitudes del frontend (como crear citas o enviar archivos), las procesa con lógica propia y responde con datos o confirma acciones. También coordina los flujos automatizados (vía webhooks) y garantiza seguridad y trazabilidad.

- **Framework Flask:** Es un microframework web para Python que permite crear aplicaciones backend ligeras, rápidas y escalables. Está diseñado para ser extensible y modular, lo que lo hace ideal para desarrollar APIs RESTful y servicios de backend modernos (Grinberg, 2018). En el sistema, Flask se utiliza para construir la API principal que gestiona usuarios, citas, autenticación y conexión con otras capas como RabbitMQ o PostgreSQL.

- **Arquitectura por capas del sistema:** Estas capas reflejan la arquitectura lógica y funcional del sistema, y deben estar definidas dentro del marco teórico, no solo en la metodología:

- **Capa de Presentación**

  Gestiona la interacción del usuario. Incluye la interfaz web de la clínica y una vista especial para Raspberry Pi en modo kiosko. Permite registrar citas, acceder a documentos, y autenticar al usuario, conectando con la capa de servicios.

- **Capa de Servicios**

  Implementada con Flask, esta capa contiene la lógica de negocio principal: gestión de workspaces, citas, archivos, y control de conexión con WhatsApp mediante Evolution API. Expone rutas API que son invocadas por la interfaz o por n8n.

- **Capa de Automatización e IA**

  Construida con n8n, aquí se definen los flujos de trabajos que responden a eventos del sistema (como carga de documentos o mensajes entrantes). Se integra con Azure OpenAI y Qdrant para realizar ingesta de conocimiento y generar respuestas conversacionales.

- **Capa de Infraestructura de Datos e Integraciones**

  Soporta la persistencia y comunicación del sistema. Contiene PostgreSQL (estructurado por esquemas por workspace), Azure Blob Storage (para documentos), RabbitMQ (para eventos) y Evolution API (para conectar con WhatsApp). Esta capa mantiene la integridad y coordinación entre todos los servicios.

- **Gestión de workspaces en sistemas multitenant:** Un workspace (espacio de trabajo) representa un entorno lógico y aislado dentro de una aplicación multitenant.
  Cada workspace puede tener su propio conjunto de datos, configuraciones y usuarios, permitiendo a varios clientes compartir la misma infraestructura sin que sus datos se mezclen (Microsoft, 2023b). En el proyecto, cada veterinaria o clínica representa un workspace. Esto permite escalar horizontalmente el sistema y mantener separación de datos mediante esquemas dedicados en PostgreSQL y almacenamiento específico por cliente.

1.9.5 Infraestructura y servicios en la nube

- **Computación en la nube:** La computación en la nube (cloud computing) es un modelo de entrega de servicios informáticos bajo demanda a través de Internet. Permite a los usuarios acceder a recursos como almacenamiento, bases de datos, servidores y servicios de inteligencia artificial sin necesidad de mantener infraestructura física propia (Mell & Grance, 2011). En el contexto del proyecto, la nube se utiliza para alojar servicios distribuidos como almacenamiento de documentos, cómputo para IA, y servicios externos como Azure OpenAI, todo ello gestionado desde entornos escalables y seguros. Su uso permite alta disponibilidad**,** elasticidad y fácil integración entre componentes como bases de datos, APIs y flujos automatizados.

- **Almacenamiento en la nube (Azure Blob Storage y PostgreSQL):**

  Azure Blob Storage es un sistema de almacenamiento masivo no estructurado de objetos en la nube de Microsoft Azure. Está diseñado para guardar documentos, imágenes, vídeos o archivos de texto a gran escala (Microsoft, 2023a). En el proyecto, se usa para almacenar archivos veterinarios, PDFs y documentos procesados, mientras que PostgreSQL se emplea como base de datos relacional para almacenar metadatos, registros de usuarios, citas y enlaces a esos archivos. Ambos servicios se integran: los archivos se guardan físicamente en Blob Storage y sus referencias (nombres, rutas, propietario) se registran en PostgreSQL, lo cual permite una separación lógica y segura de los datos y su estructura.

- **Servicios de inteligencia artificial en la nube (Azure OpenAI):**
  Azure OpenAI Service ofrece acceso a modelos de lenguaje como GPT directamente desde la infraestructura de Azure. A través de este servicio, se pueden generar textos, responder preguntas o crear embeddings para búsquedas semánticas, todo con integración segura en entornos empresariales (Microsoft, 2023c). En el proyecto, se usa para generar embeddings que alimentan la base vectorial y para generar respuestas contextuales en conversaciones con usuarios.

1.9.6 Integración con plataformas de mensajería

- **Evolution API (integración con WhatsApp):** Evolution API es una solución de mensajería que permite integrar sistemas propios con WhatsApp Business a través de una API. Soporta funciones como envío y recepción de mensajes, envío de medios y manejo de sesiones de usuarios (EvolutionAPI, 2023). En el sistema, se utiliza para recibir mensajes de clientes (que pueden ser audios, imágenes o texto) y activar flujos automatizados en n8n, conectando el canal de WhatsApp con el backend del sistema y la IA conversacional.

1.9.7 Despliegue y gestión de servicios

- **Contenerización con Docker y Gestión de contenedores con Portainer:** Para operar una solución compuesta por varios servicios (por ejemplo, chatbot, automatización y almacenamiento), es conveniente usar tecnologías de contenedores.

  Docker permite empaquetar cada componente con sus dependencias en contenedores, asegurando una ejecución consistente en distintos entornos y facilitando el despliegue y la actualización de servicios de forma independiente (Oracle, 2021). Además, al ser más livianos que las máquinas virtuales, los contenedores simplifican el escalamiento y reducen la sobrecarga operativa.

  Para administrar múltiples contenedores, Portainer ofrece una interfaz gráfica que centraliza tareas de monitoreo y gestión (estado de servicios, logs, redes y permisos), reduciendo la complejidad de la administración por consola y ayudando a mantener el sistema de forma más ordenada en operación (Bin, 2024). En conjunto, Docker aporta portabilidad y consistencia, y Portainer facilita el control operativo del despliegue.

1.9.8 Gestión del contexto en sistemas conversacionales

- **Contexto conversacional o buffer de contexto**: La gestión del contexto es un componente clave en aplicaciones de IA conversacional, ya que permite que el chatbot mantenga coherencia y continuidad durante el diálogo.

  Esta “memoria” funciona como un buffer que conserva información previa (intenciones, preferencias y datos proporcionados) para evitar preguntas repetidas y

ofrecer respuestas más relevantes y personalizadas; por ejemplo, recordar el nombre de la mascota o una fecha preferida ya mencionada (Q2BStudio, 2025)

En términos generales, el contexto puede manejarse de forma temporal dentro de la sesión (manteniendo un historial reciente de mensajes) o mediante mecanismos externos cuando se requiere sostener conversaciones largas o persistir información entre sesiones.

Entre las estrategias más comunes están el uso de ventanas de historial, resúmenes automáticos de lo conversado y almacenamiento de datos clave para recuperarlos cuando sea necesario (Q2BStudio, 2025).

1.9.9 Interacción multimodal en inteligencia artificial

- **Procesamiento multimodal (voz, imagen y texto):** La interacción multimodal en IA se refiere a la habilidad de un sistema para analizar e integrar diferentes modos de comunicación principalmente texto, voz e imágenes de forma unificada, alcanzando una comprensión más integral del contexto que los métodos convencionales unidimensionales (Cole, 2024; Fernández, 2025).

  En la práctica, esto permite que un asistente "lea" texto, "escuche" mensajes de voz y "analice" contenido visual; por ejemplo, combinar la descripción de síntomas con una fotografía para responder con mayor precisión.

  En este proyecto, la multimodalidad se refleja en tres vías:

  (1) voz, mediante reconocimiento automático del habla (ASR) para transcribir audios a texto y síntesis de voz (TTS) para responder en audio,

  (2) imágenes, permitiendo procesar fotos o documentos (p. ej., lesiones o registros) usando visión por computadora y, cuando corresponda, OCR para extraer información,

  (3) texto, como modalidad base para la interacción y el procesamiento del modelo (Fernández, 2025).

Al habilitar estos canales (por ejemplo, WhatsApp con audio o fotos, o chat web), el sistema no se limita a un chat escrito, sino que se adapta al medio preferido del usuario, mejorando accesibilidad, comodidad y naturalidad en la atención, a la vez que fortalece la automatización del servicio (Cole, 2024).

1.9.10 Validación y pruebas del sistema

- **Pruebas End-to-End (E2E):**

  Las pruebas end-to-end (E2E) son una metodología de validación que verifica que todo un sistema funciona como se espera, simulando el flujo real del usuario desde la interfaz hasta el backend y las integraciones externas (Umar, 2021). Este tipo de pruebas comprueba desde el inicio (por ejemplo, agendar una cita desde la interfaz) hasta el final (guardar en la base de datos, enviar confirmación y actualizar calendarios). En el proyecto, las pruebas E2E ayudan a validar flujos automatizados como agendamientos vía chatbot o la recepción de documentos que desencadenan procesos en n8n.

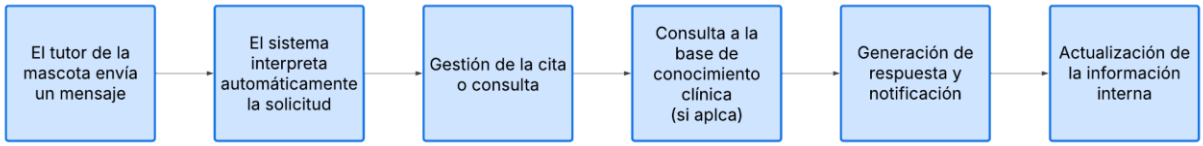
Capítulo 2

2.1 Metodología de desarrollo

Para el desarrollo de la plataforma integral de gestión y soporte al cliente en clínicas veterinarias, se adoptó el Marco de Automatización Inteligente (Intelligent Automation Framework) como metodología de desarrollo (Williams & Olajide, 2022), que integra de forma estructurada tres ejes principales: a) la automatización de procesos mediante flujos de trabajos (workflow) orquestados en n8n, b) los servicios de inteligencia artificial (IA) para análisis y toma de decisiones y, c) los elementos de Internet de las Cosas (IoT) para la visualización local de información relevante en la clínica veterinaria.

Figura 2.1

Flujo general del proceso automatizado de atención veterinaria



El proceso específico que se desea automatizar en la **Figura 2.1** corresponde al ciclo de atención de citas veterinarias, desde la comunicación inicial del tutor de la mascota (usando un canal de WhatsApp), pasando por la interpretación automática del mensaje, la gestión de la cita, el uso de una base de conocimiento clínica, y la notificación tanto al personal de la clínica como al tutor de la mascota.

Adicionalmente, la plataforma incorpora flujos de trabajos para la gestión y actualización de la base de conocimiento a partir de archivos cargados por el personal de la clínica y la administración del repositorio documental en línea.

A continuación, se describen las fases del marco de automatización inteligente empleadas en este proyecto.

2.2 Fase 1: Descubrimiento del proceso

En esta etapa se identificó y describió el proceso real de la clínica veterinaria para la gestión de citas y administración de información y documentos, con el objetivo de comprender cómo se ejecuta actualmente, definir el proceso propuesto con la solución, y establecer el plan de cambios y tareas necesarias para la transición.

La caracterización del proceso se realizó mediante entrevistas con el personal de la clínica. Se realizaron 3 entrevistas semiestructuradas con 2 participantes (personal responsable de la atención y gestión de citas), en modalidad virtual, con una duración aproximada entre 40 minutos cada una.

El personal entrevistado cuenta con varios años de experiencia y participan directamente en el registro de citas, comunicación con los tutores de mascota y gestión de documentos clínicos.

La información se recolectó mediante una guía de preguntas orientada a identificar actividades, responsables, entradas/salidas y puntos de dolor del proceso. Además, se tuvo una observación del flujo operativo típico y revisión de los puntos de interacción con los tutores de mascotas. El resultado se consolidó en mapas de proceso para evidenciar brechas (ineficiencias, duplicidad, pérdidas de información y carga operativa).

2.2.1 Proceso actual

La fase de descubrimiento del proceso tuvo como objetivo comprender en profundidad la forma en que la clínica veterinaria gestiona actualmente las citas y la información asociada, así como identificar oportunidades de mejora mediante la automatización inteligente.

En términos de gestión por procesos (BPM), AS-IS describe el proceso actual tal como se ejecuta, TO-BE representa el proceso objetivo propuesto tras la mejora, y TO-DO corresponde al conjunto de acciones y actividades necesarias para realizar la transición del AS-IS al TO-BE (SYDLE, 2021).

En el proceso actual (AS-IS), las solicitudes de citas se reciben principalmente a través de mensajes enviados por los tutores de las mascotas.

Dichos mensajes presentan un alto grado de variabilidad en su redacción, lo que obliga al personal de la clínica a interpretar manualmente la intención del tutor de la mascota, revisar

la disponibilidad de horarios y registrar la cita de forma manual y en ocasiones asistida por sistema.

Este procedimiento incrementa la carga operativa y genera dependencia del conocimiento tácito del personal. De manera similar, los documentos clínicos y guías de atención se gestionan de forma dispersa, ya sea en archivos individuales, conversaciones previas o repositorios no estandarizados, dificultando su consulta rápida y consistente.

A partir de este análisis, se definió un proceso objetivo (TO-BE) orientado a la centralización y automatización de la gestión de citas y de la información clínica.

En este proceso propuesto, la interacción del personal de la clínica se realiza a través de una interfaz dedicada para el registro y visualización de citas, así como para la carga de documentos que alimentan una base de conocimiento.

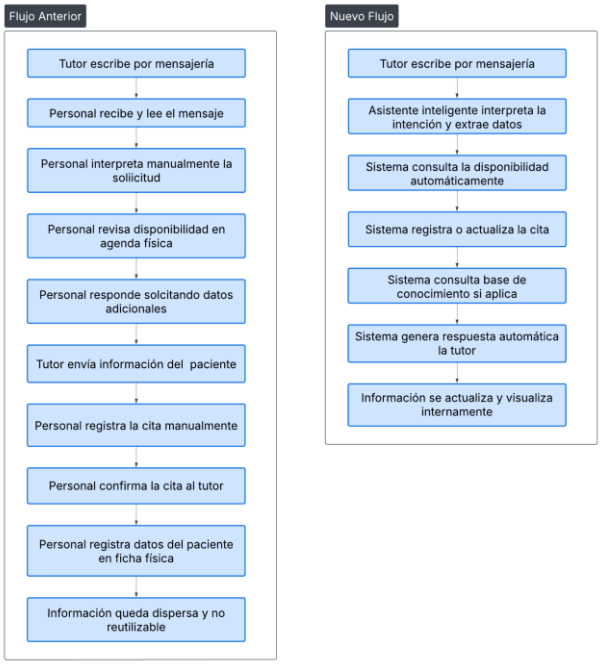
La automatización del flujo se apoya en un orquestador de flujos de trabajos que coordina la clasificación de solicitudes, la persistencia de datos y la consulta de información, reduciendo significativamente la intervención manual y poder mejorar la trazabilidad del proceso.

La comparación entre el proceso actual y el proceso propuesto permitió establecer un plan de transformación (TO-DO), que contempla la definición de reglas de negocio, la estructuración de la información clínica y la validación del nuevo flujo mediante pruebas de usabilidad.

En la **Figura 2.2** se presenta un mapa comparativo de los procesos AS-IS y TO-BE, donde se logra apreciar de forma clara la transición desde un flujo predominantemente manual hacia un proceso automatizado, centralizado y soportado por inteligencia artificial.

Figura 2.2

Flujo anterior y propuesto del proceso automatizado.



2.3 Fase 2: Diseño del flujo automatizado en n8n

Una vez definido el proceso objetivo (TO-BE), se procedió a diseñar el flujo automatizado, utilizando n8n como orquestador principal.

La selección de n8n como orquestador se basó en criterios técnicos y operativos: (i) disponibilidad de conectores y soporte nativo para Webhooks/HTTP, (ii) facilidad de despliegue en contenedores Docker y administración en Portainer, (iii) capacidades de trazabilidad y monitoreo de ejecuciones, (iv) manejo de credenciales y variables de entorno para integrar servicios externos, y (v) rapidez de iteración para construir y mantener flujos de trabajos sin incrementar complejidad del backend. (Oriigin, 2025).

Tabla 2.1

Comparación técnica breve de orquestadores de flujos de trabajo.

Criterio	n8n	Node-RED	Apache Airflow	Zapier / Make
Modelo de despliegue	**Self-host** (Docker) o cloud; control de datos	Principalmente **edge/IoT** o servidor propio; ligero	Self-host (infraestructura más pesada); orientado a clúster	Principalmente **SaaS** (dependencia del proveedor)

Enfoque de ejecución	**Event-driven** (webhooks, APIs) + tareas	**Event-driven** (ideal para eventos/IoT)	**Batch/ scheduler** (DAGs)	Event-driven, pero con límites por plan y plataforma
Integraciones y conectores	Amplio catálogo + custom HTTP; plantillas	Muchísimos nodos (ecosistema IoT fuerte)	Integraciones vía "operators/ providers" (más código)	Muchas integraciones, pero atadas al marketplace/ planes
Extensibilidad	Visual + scripts; integraciones por API	Muy extensible vía nodos y funciones JS	Extensible en Python, pero **más ingeniería**	Extensión limitada a lo que permite la plataforma
Trazabilidad/o peración	UI para ejecuciones; útil para soporte	UI y depuración de flows; muy práctico	UI avanzada para monitoreo de DAGs	Logs/ observabilidad dependen del plan
Ajuste al caso (WhatsApp/ API + BD + IA + self-host)	Alto: integra APIs/IA y se self-hostea	Medio: fuerte en IoT; integraciones "enterprise" varían	Medio-bajo: mejor para procesos por lote	Medio: rápido, pero trade-off en control/lock-in

Nota. Elaboración propia a partir de (n8n, Node-RED, Airflow, Zapier y Make, 2026)

Con base en la comparación de alternativas mostradas en la **Tabla 2.1**, se seleccionó n8n porque ofrece un equilibrio adecuado entre despliegue autoalojado (control de datos), orquestación orientada a eventos mediante webhooks/APIs y facilidad de integración con servicios externos (bases de datos e IA).

En contraste, Node-RED se ajusta mejor a escenarios centrados en IoT/edge, Airflow se orienta a orquestación batch, y las opciones SaaS (Zapier/Make) incrementan la dependencia del proveedor y limitan el control del entorno de despliegue (Node-RED, s. f.; n8n, Apache Software Foundation, 2026.; Zapier, s. f.; Make, s. f.).

Esta etapa de diseño se buscó traducir el proceso conceptual propuesto en una secuencia operativa de eventos, decisiones y acciones automatizadas, manteniendo una separación clara entre la lógica de negocio, los servicios de inteligencia artificial y los mecanismos de persistencia de datos.

El diseño del flujo automatizado se estructuró a partir de la identificación de los eventos clave que intervienen en la operación del sistema, tales como la creación de una cita, la carga de documentos clínicos y la recepción de consultas por parte de los tutores de las mascotas.

Cada uno de estos eventos activa un conjunto de acciones coordinadas por el orquestador, que incluyen validaciones de acceso, procesamiento de información y comunicación con servicios externos. De este modo, n8n actúa como el eje central que asegura la coherencia del proceso y la correcta propagación de la información entre los distintos componentes del sistema.

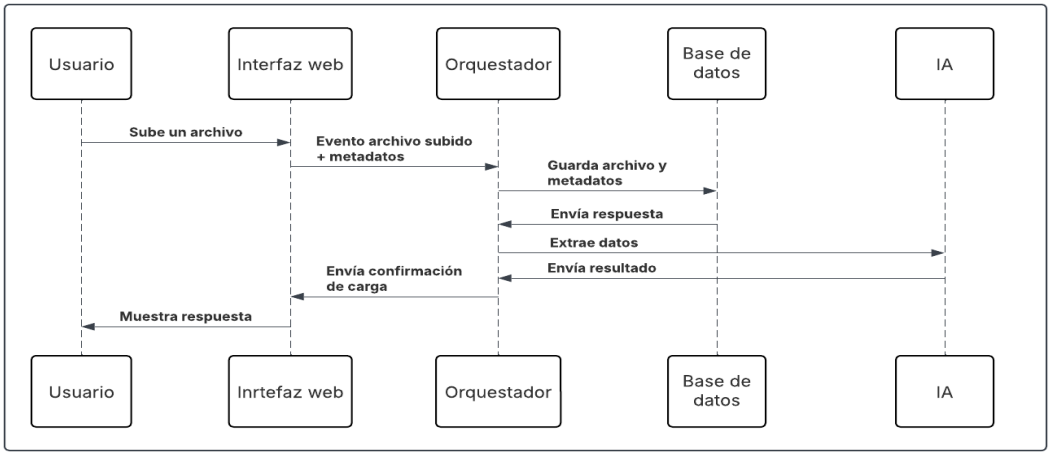
En particular, el flujo de carga de documentos fue diseñado para garantizar tanto la persistencia segura de la información como su aprovechamiento en la base de conocimiento.

Cuando la clínica carga un archivo, el sistema valida la identidad del usuario, registra los metadatos asociados y almacena el documento en un repositorio en línea. Posteriormente, el contenido es procesado para su integración en un sistema de recuperación de información asistido por IA, lo que permite que futuras consultas se responden utilizando información oficial proporcionada por las doctoras.

Este flujo se ilustra en la **Figura 2.3**, donde se representa la secuencia de interacciones entre la interfaz, el orquestador, los servicios de almacenamiento y los componentes de inteligencia artificial.

Figura 2.3

Secuencia de interacciones entre los componentes del sistema.

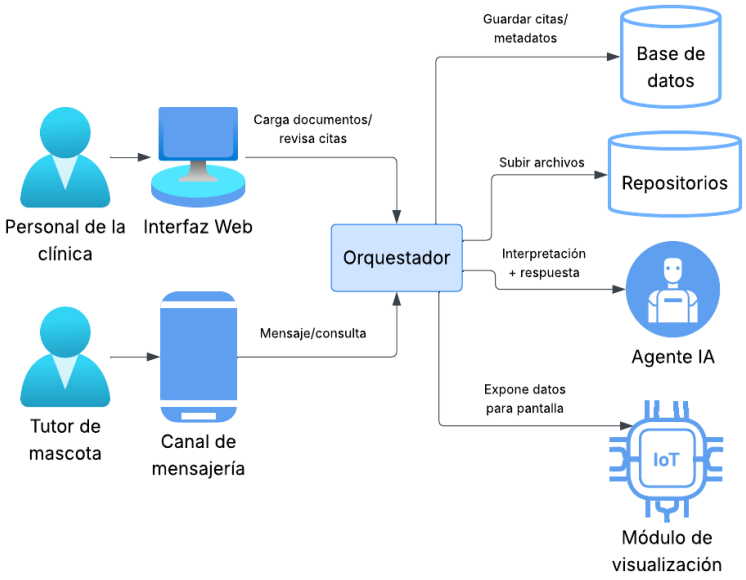


De forma complementaria, el diseño general del sistema contempla la integración de los distintos flujos automatizados dentro de una arquitectura unificada, en la cual n8n coordina las interacciones con las bases de datos, los servicios de IA y el componente IoT encargado de la visualización local de citas y recordatorios.

La **Figura 2.4** muestra esta arquitectura de orquestación, permitiendo comprender cómo los distintos módulos se articulan para soportar el proceso propuesto.

Figura 2.4

Arquitectura de orquestación de componentes del sistema



La solución se diseña siguiendo una arquitectura por capas, la cual se implementa y valida en las fases posteriores del proyecto.

2.4 Fase 3: Visualización de información

El componente IoT del sistema consiste en un módulo de visualización compuesto por una Raspberry Pi y una pantalla dedicada, permitiendo la presentación de información en tiempo real sin necesidad de sensores externos. El componente, ubicado en la clínica, muestra las citas programadas, recordatorios importantes, y otra información relevante para el personal de la clínica.

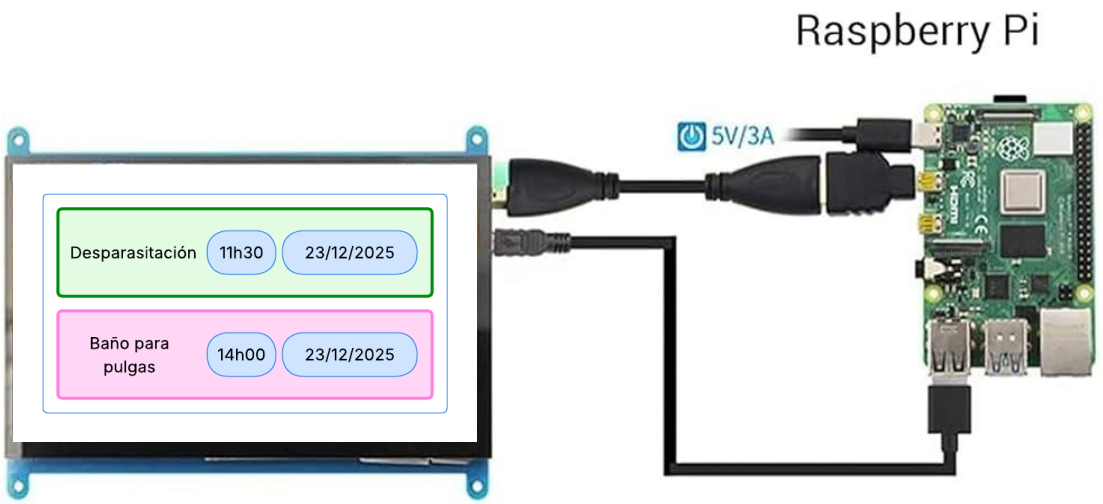
La Raspberry actúa como “nodo de visualización local” que consume información generada por los flujos de trabajos en n8n y la presenta de forma clara al personal de la clínica.

Esto se logra por medio de una comunicación lógica en donde n8n actualiza la información de citas y recordatorios en la base de datos y en un endpoint consumido por sí mismo.

Luego se especificó la visualización de los datos para la pantalla donde se incluye formatos de las citas a mostrar (fecha, hora, motivo de la cita) en la estructura definida en la **Figura 2.5**.

Figura 2.5

Previsualización de la pantalla de las citas registradas.



Se estableció que los flujos de trabajos de n8n sean los responsables de disparar la actualización de la información que la Raspberry consulta o recibe, garantizando coherencia con el resto del sistema.

En esta etapa se asegura que la solución no solo automatice la lógica de negocio y la IA, sino que además entregue información accionable en el entorno físico de la clínica.

2.5 Fase 4: Inserción de IA

Esta etapa define e implementa los puntos de integración de IA dentro de los flujos de trabajos orquestados por n8n, con el objetivo de automatizar la interpretación de mensajes, estructurar la información de citas y soportar consultas informativas basadas en documentos de la clínica.

La inserción de IA se diseñó de forma modular: cada workflow invoca servicios externos de IA mediante endpoints, y consume sus salidas como insumo para decisiones del flujo, manteniendo separadas la orquestación, la lógica de negocio y la persistencia. En particular, se implementaron los siguientes módulos:

- **Clasificación del mensaje (intención):** ante un mensaje entrante, el workflow solicita a un modelo de lenguaje una etiqueta de intención (p. ej., nueva cita, reprogramación/cancelación, consulta informativa u otros). Esta etiqueta determina la rama del flujo, incluyendo reglas de alternativas cuando la clasificación es ambigua.

- **Extracción de entidades para gestión de citas:** para intenciones asociadas a citas, se extraen campos estructurados (tutor, mascota, fecha/hora, servicio). El workflow valida campos mínimos (p. ej., fecha e identificación del tutor) y, si falta información, genera una pregunta de aclaración antes de registrar la cita o ejecutar cambios.

- **Respuesta informativa con recuperación documental:** para consultas clínicas, el workflow ejecuta un subflujo de recuperación–generación: (i) transforma la pregunta en una representación vectorial, (ii) consulta un índice vectorial en Qdrant para recuperar fragmentos relevantes de documentos cargados por la clínica, y (iii) genera una respuesta basada en dichos fragmentos, priorizando consistencia con la documentación interna.

- **Generación de respuesta y trazabilidad:** el resultado final se envía al canal conversacional y se registran metadatos por ejecución (intención detectada, entidades extraídas, identificadores del workflow, tiempos y estados). Esto permite auditar fallos, ajustar prompts y refinar reglas de validación sin alterar el proceso clínico.

- **Integración de servicios:** los modelos se consumen mediante API (p. ej., OpenAI), y el workflow gestiona reintentos y manejo de errores ante tiempos de espera o fallos de servicios externos, garantizando que el proceso pueda degradar de forma controlada (p. ej., solicitar aclaración o derivar al personal de la clínica).

2.6 Fase 5: Pruebas de extremo a extremo (E2E)

Esta fase se orienta a la validación práctica del funcionamiento integrado del sistema, considerando el recorrido completo de la información desde la interacción inicial del tutor de la mascota hasta la visualización y gestión de las citas por parte del personal de la clínica. El

objetivo de esta fase es verificar que los componentes desarrollados operen de manera coordinada en un contexto cercano al uso real.

Para cada escenario evaluado se definieron criterios simples y observables, acordes a las capacidades del sistema y a los datos efectivamente obtenidos durante las sesiones de prueba. Los criterios de evaluación considerados fueron: (i) cumplimiento del flujo completo de la tarea, (ii) tiempo aproximado de respuesta del sistema, (iii) registro correcto de la información en la plataforma, y (iv) observaciones cualitativas sobre errores, dificultades o comportamientos inesperados.

Como parte de la validación funcional, se definió el ***indicador de automatización de la gestión de citas***, el cual permite cuantificar el grado en que las solicitudes relacionadas con citas son resueltas sin intervención humana. Este indicador se calcula como la relación entre el número de solicitudes de gestión de citas completadas automáticamente por el sistema y el total de solicitudes evaluadas durante la sesión de prueba, tal como se indica en la ecuación 2.1.

$$Porcentaje\ de\ automatización\ de\ citas = \left(\frac{Solicitudes\ de\ citas\ resueltas\ automáticamente}{Total\ de\ solicitudes\ de\ gestión\ de\ citas}\right) \times 100 \quad (2.1)$$

Metodológicamente se definieron los siguientes escenarios, para validar que el sistema **cumple los niveles de desempeño esperados** antes de producción:

2.6.1 Pruebas funcionales E2E (integración):

- **Caso agendamiento de cita:** En este escenario se verificó el correcto funcionamiento del flujo de agendamiento de citas, comprobando que el sistema complete la secuencia esperada desde la solicitud inicial hasta el registro final de la cita en la plataforma. La validación incluyó la confirmación visual de que la cita se almacena correctamente sin duplicaciones y la observación del comportamiento del asistente conversacional durante la interacción. Adicionalmente, se recopilaron observaciones cualitativas del personal de la clínica sobre la claridad del flujo y la interacción con el asistente.

- **Caso consulta informativa apoyada en RAG:** En este escenario, el personal de la clínica carga documentos informativos en el sistema, los cuales posteriormente son utilizados para responder consultas realizadas por los tutores de las mascotas. El objetivo de esta prueba fue verificar que el sistema pueda recuperar información relevante y responder de manera coherente con el contenido disponible.

Las métricas consideradas fueron el cumplimiento de la consulta, el tiempo aproximado de respuesta y la evaluación cualitativa de la coherencia de la respuesta, basada en la percepción del personal de la clínica sobre si la información entregada corresponde al contenido cargado.

2.6.2 Pruebas de usabilidad (UX):

Las pruebas de usabilidad se realizaron mediante una sesión práctica con personal de la clínica, durante la cual se interactuó con el asistente virtual de WhatsApp y la plataforma web para ejecutar tareas reales como la revisión de citas, confirmaciones y consultas informativas. Durante esta sesión se evaluaron aspectos relacionados con la facilidad de uso, claridad de los textos, comprensión de los botones y organización visual de la interfaz.

La evaluación se apoyó en la observación directa del uso del sistema, el registro de tiempos aproximados para completar las tareas y la recopilación de comentarios cualitativos del personal de la clínica. Adicionalmente, se utilizó una escala de percepción de facilidad para cada tarea, permitiendo identificar puntos de fricción y oportunidades de mejora en la interacción.

2.7 Fase 6: Despliegue

A nivel metodológico, el despliegue se concibe como:

2.7.1 Definición de servicios y dependencias en contenedores: Separa cada componente en un contenedor independiente para facilitar escalabilidad y mantenimiento y la declaración de redes internas entre contenedores para asegurar comunicación controlada.

2.7.2 Uso de Portainer para gestión operativa: Monitorea el estado de los contenedores y permite el arranque, reinicio y actualización de servicios sin afectar al resto de la infraestructura.

2.7.3 Separación de entornos (desarrollo / pruebas / producción): Aunque en una primera fase puedan compartir infraestructura, se establece el criterio metodológico de separar configuraciones (variables de entorno, claves, endpoints) para reducir riesgos en el despliegue a producción.

2.8 Fase 7: Monitorización continua y mejora iterativa

En esta fase se establece la estrategia de monitorización del sistema durante el periodo de validación, con el objetivo de verificar su estabilidad operativa y recopilar información que permita realizar ajustes incrementales. La monitorización se orienta a evaluar el desempeño real en condiciones operativas.

Métricas de desempeño

Para evaluar la viabilidad del sistema durante el periodo de validación, se definieron las siguientes métricas de desempeño:

2.8.1 Disponibilidad del servicio (uptime):

Esta métrica se define como el porcentaje de tiempo en el que la plataforma web y el asistente conversacional se mantienen operativos y accesibles durante su operación en producción. Su cálculo se basa en la relación entre el tiempo total de funcionamiento del sistema y el tiempo total de observación, tal como se indica en la ecuación 2.2.

$$Disponibilidad = \frac{Tiempo\ operativo}{Tiempo\ total} \times 100 \quad (2.2)$$

2.8.2 Tasa de éxito de ingestión de documentos:

La ecuación 2.3 representa la relación porcentual entre los documentos que fueron correctamente procesados e indexados en la base de conocimiento y el total de documentos cargados por el personal de la clínica.

$$Tasa\ de\ ingestión = \frac{Documentos\ indexados\ correctamente}{Total\ de\ documentos\ cargarg\ a\ dos} \times 100 \quad (2.3)$$

Adicionalmente, los documentos que no pudieron ser procesados se registran para identificar causas comunes de fallo, como formato incompatible, tamaño del archivo o errores durante la comunicación con los servicios de procesamiento.

2.8.3 Retroalimentación del personal de la clínica

Como complemento a los indicadores técnicos, se incorporó la retroalimentación del personal de la clínica como insumo para la mejora del sistema. Esta retroalimentación se recopiló durante las sesiones de prueba mediante comentarios cualitativos sobre la claridad de la información presentada en la visualización de citas, la utilidad del calendario para la organización diaria, la calidad de las respuestas generadas por el asistente conversacional y la facilidad de uso de la plataforma para cargar documentos y gestionar citas.

La información obtenida se recopiló mediante un cuestionario cualitativo aplicado al personal de la clínica, el cual se incluye como Apéndice A. Esto permitió identificar oportunidades de mejora tanto en la interacción con el asistente conversacional como en la presentación visual de la información de la agenda.

Capítulo 3

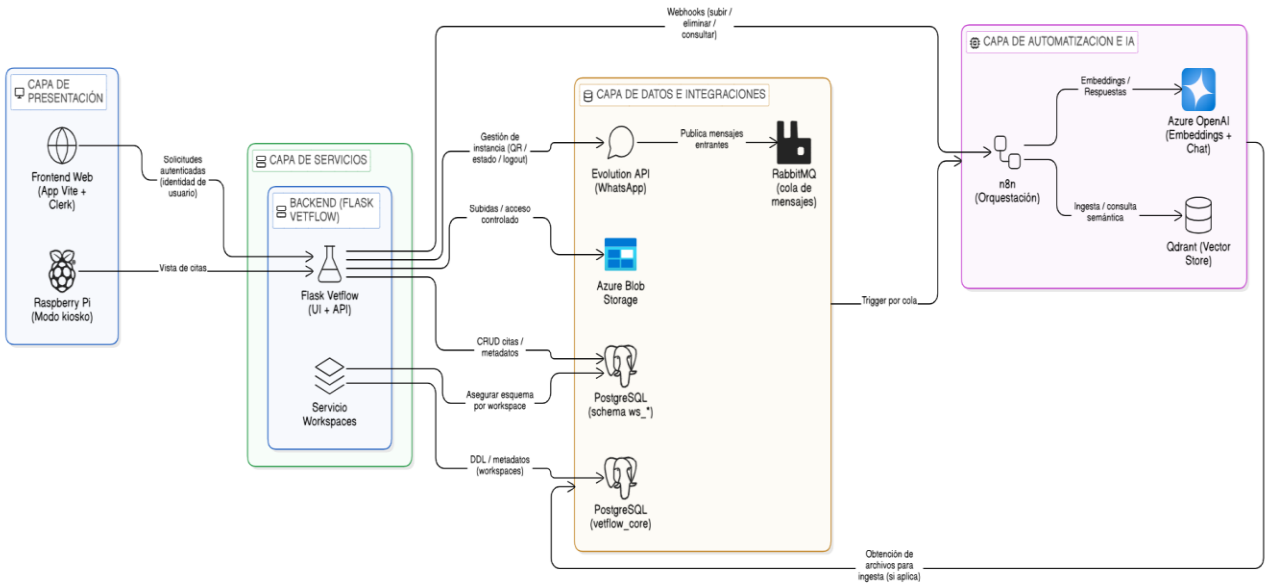
3.1 Diseño e Implementación

3.1.1 Diseño de la solución

El diseño de la solución se materializa mediante una arquitectura por capas que organiza los componentes del sistema y define su interacción. Como se observa en la **Figura 3.1**, la capa de presentación permite al personal de la clínica acceder a la gestión de citas y a la visualización de información mediante una interfaz web, la cual puede ser consultada desde distintos dispositivos, incluido un navegador en una Raspberry Pi. La capa de servicios centraliza la lógica de negocio y actúa como punto de acceso a las funcionalidades del sistema. La capa de automatización e inteligencia artificial gestiona los flujos de atención de mensajes y la ingesta de documentos, mientras que la capa de datos e integraciones soporta la persistencia, la mensajería y la conexión con el canal WhatsApp. Esta organización facilita el desacoplamiento de responsabilidades y la evolución del sistema.

Figura 3.1

Arquitectura por capas de la plataforma y servicios integrados.



3.1.2 Implementación de la solución

La implementación se realizó conectando progresivamente las capas definidas en el diseño, priorizando primero la operación interna (interfaz, workspaces, citas y archivos), luego la automatización del RAG, y finalmente la integración del canal WhatsApp y la visualización operativa en pantalla.

- **Operación interna: Manejo de la interfaz**

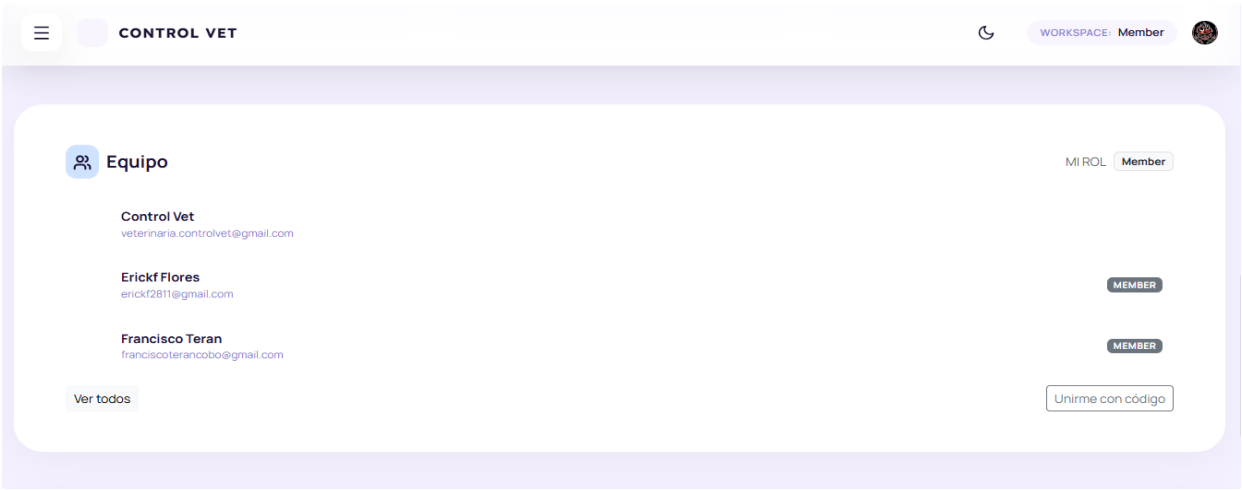
La operación interna del sistema se apoya en una separación clara entre la interfaz web y la lógica de procesamiento. La interfaz es utilizada por el personal de la clínica para gestionar citas, documentos y la conexión con el canal WhatsApp, actuando como un punto de interacción y control.

El backend, implementado con Flask, funciona como el componente central que recibe las solicitudes de la interfaz, valida el contexto activo y coordina las operaciones con los servicios de persistencia y automatización. De esta manera, la interfaz no ejecuta lógica de negocio directamente, sino que delega las acciones relevantes al backend.

El manejo de los espacios de trabajo (workspaces), mostrado en la **Figura 3.2**, se integra en esta capa para garantizar que cada operación se realice sobre el conjunto de datos correspondiente. Al seleccionar un workspace, el sistema utiliza automáticamente el esquema asociado, asegurando el aislamiento lógico de la información sin que el usuario deba realizar configuraciones adicionales.

Figura 3.2

Centro de administración del personal de la clínica dentro de un workspace



- **Gestión de citas y archivos**

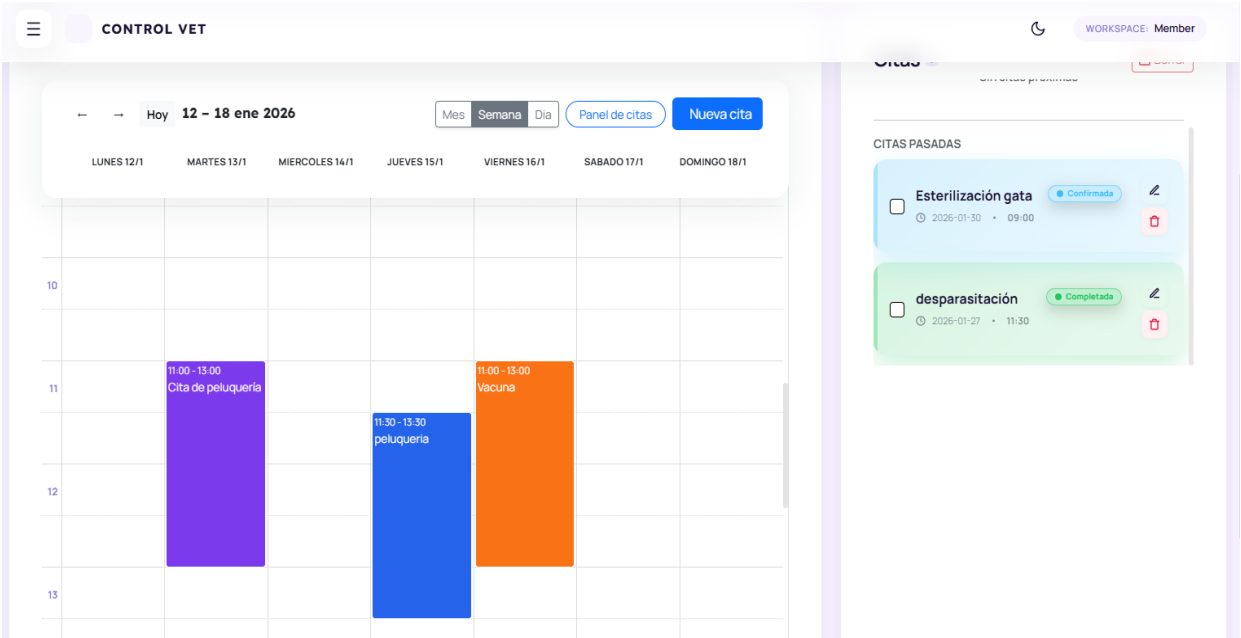
La creación o selección de un workspace activa la verificación del esquema correspondiente, lo que permite que la gestión de citas y archivos funcione de forma uniforme para distintos espacios sin que el usuario perciba cambios técnicos. Con este enfoque, las operaciones diarias (por ejemplo, registrar una cita o listar citas pendientes) se realizan siempre contra el esquema asociado al workspace activo.

La gestión de citas, apreciado en la **Figura 3.3**, se realiza mediante una vista de calendario interactivo dentro de la interfaz web, donde el personal de la clínica puede visualizar, arrastrar y reprogramar citas de forma directa. Cada cita se asocia a un estado operativo que permite organizar el flujo de atención y las acciones automáticas del sistema.

Los estados definidos para una cita son: programada, por confirmar, completada, cancelada y no asistió. El estado **"programada"** indica que la cita ha sido registrada; **"por confirmar"** señala que se encuentra pendiente de validación; **"completada"** corresponde a una cita atendida; **"cancelada"** a una cita anulada; y **"no asistió"** a los casos en que el tutor no se presentó.

Figura 3.3

Calendario y panel con citas registradas.



El cambio de estado de una cita genera notificaciones automáticas al tutor asociado, y adicionalmente el sistema envía recordatorios previos a la atención cuando faltan tres, dos y una hora para la cita programada.

La gestión de documentos se implementó separando el almacenamiento físico del archivo y su representación operativa dentro del sistema. Los documentos se almacenan en Azure Blob Storage, mientras que en PostgreSQL se mantienen los metadatos necesarios para administrarlos desde la interfaz y para controlar su ciclo de vida.

Desde la interfaz web, las acciones de cargar, editar o eliminar documentos disparan webhooks que activan flujos de trabajos en n8n. Esta decisión asegura que cada acción relevante del personal de la clínica tenga un efecto automático y rastreable, evitando pasos manuales fuera del sistema.

En particular, cuando se carga un documento o se solicita que este sea utilizado por el asistente conversacional basado en IA, se genera un evento que inicia la ingesta hacia el repositorio vectorial; cuando se elimina un documento, se activa el flujo correspondiente para retirar o invalidar su representación en la capa de automatización e IA.

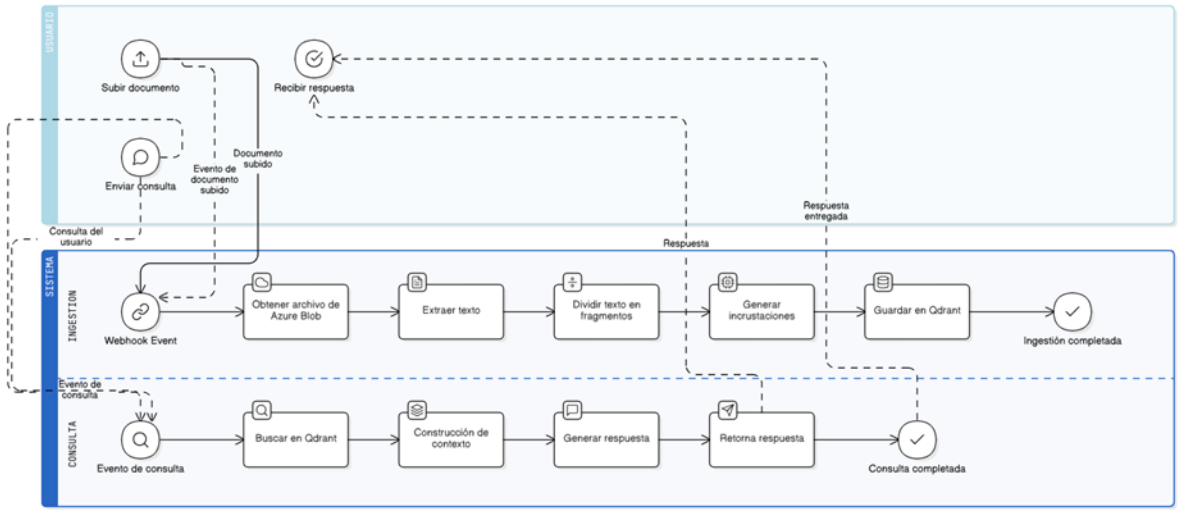
- **Automatización del RAG**

En la ingesta de información al sistema RAG se implementó en n8n utilizando Qdrant como repositorio vectorial y Azure OpenAI como proveedor de embeddings y modelo conversacional.

La **Figura 3.4** representa la implementación del flujo de ingesta. Al recibir el webhook del backend, el workflow obtiene el archivo desde el almacenamiento, extrae el contenido textual y lo prepara para indexación. Posteriormente, el contenido se segmenta y se generan embeddings que son almacenados en Qdrant, junto con la metadata necesaria para asociar cada fragmento al documento y al workspace.

Figura 3.4

Workflow de ingesta RAG hacia Qdrant usando embeddings de Azure OpenAI



De forma complementaria, la consulta se implementa de manera que, ante una conversación, el agente recupere primero fragmentos relevantes desde Qdrant y luego construya la respuesta final con el modelo conversacional. Esta implementación permite que el asistente responda con base en documentación real administrada por la clínica, manteniendo trazabilidad sobre la fuente documental sin mezclar la lógica del chat dentro de la interfaz web.

- **Integración del canal WhatsApp**

La integración con WhatsApp se implementó con Evolution API, dividiendo el problema en dos partes: *vinculación de la instancia* y *procesamiento de mensajes*.

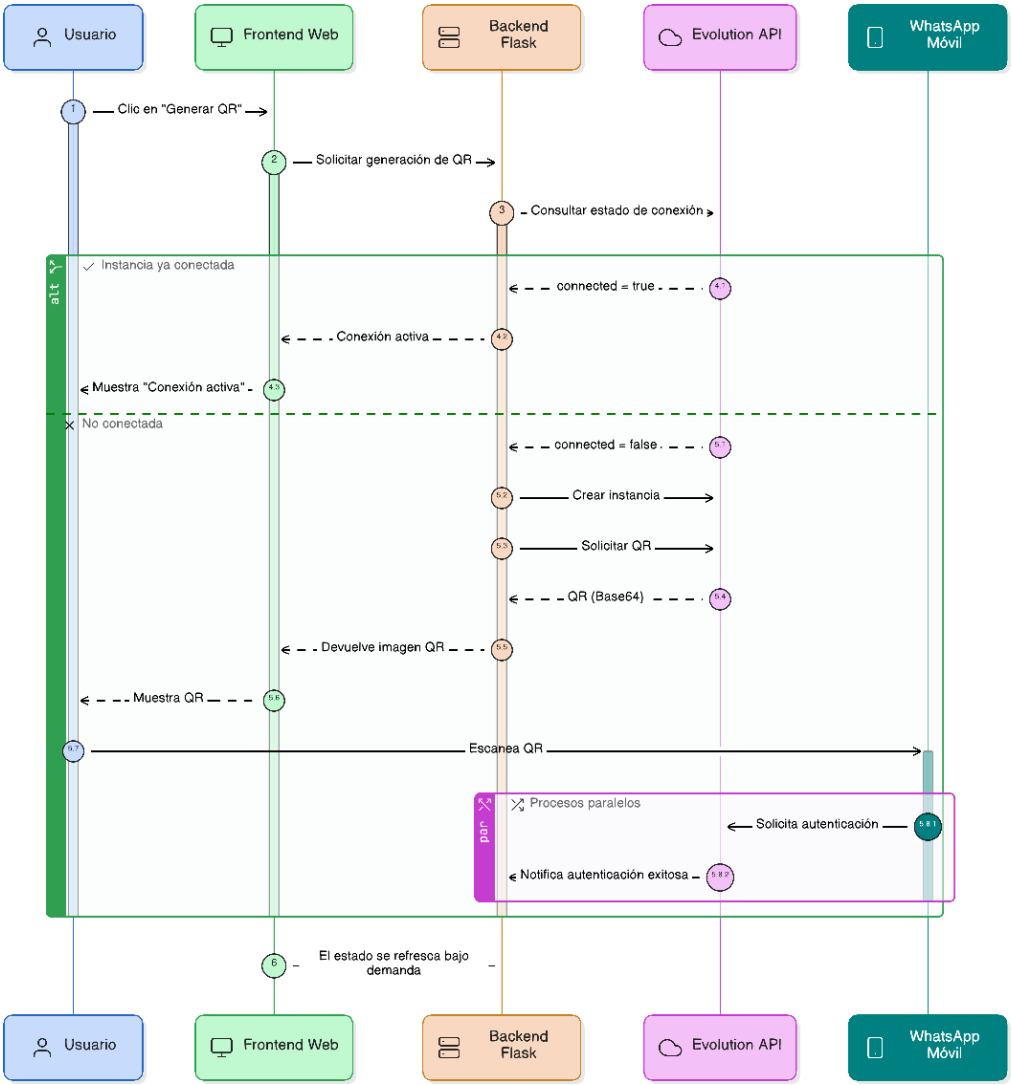
La *vinculación de la instancia* se resolvió desde la interfaz web mediante una acción "Generar QR", cuya secuencia se muestra en la **Figura 3.5**.

Cuando el usuario solicita el QR, el backend consulta el estado de conexión de la instancia en Evolution API. Si la instancia ya está conectada, el backend retorna un estado de conexión activa para evitar recreaciones innecesarias. Si no está conectada, el backend solicita la creación de instancia y luego solicita el QR para que el usuario pueda escanearlo desde el dispositivo móvil.

Esta implementación se mantuvo deliberadamente simple en la interfaz web: no se realiza una revisión constante, sino que el estado se actualiza bajo demanda cuando se abre la sección o cuando el usuario lo solicita explícitamente, reduciendo complejidad en la interfaz.

Figura 3.5

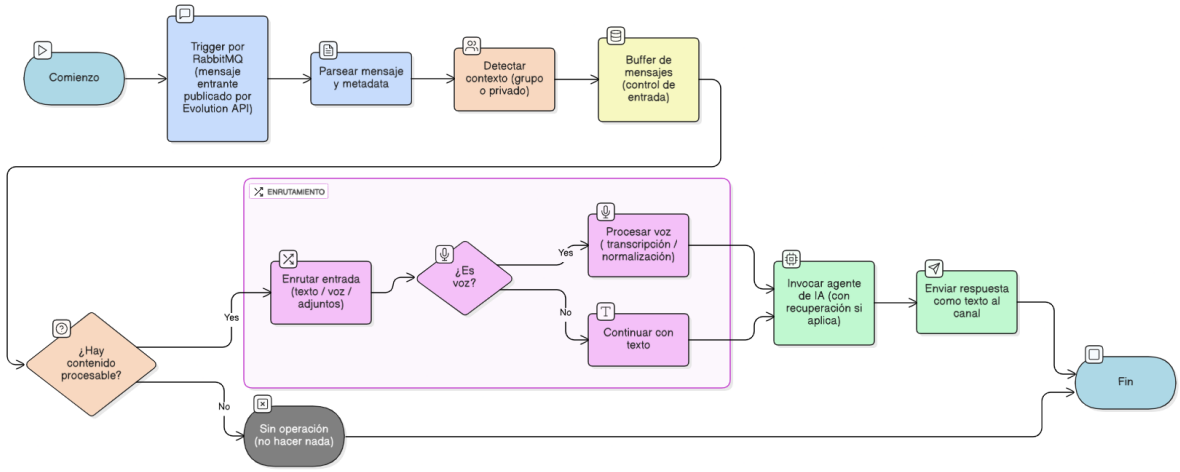
Workflow de vinculación de la instancia por medio de QR.



Mientras que, el procesamiento *de mensajes* entrantes se implementó de forma desacoplada mediante RabbitMQ, sin servicios intermedios. Evolution API publica directamente los mensajes entrantes en la cola, y n8n consume esos eventos con un disparador que genere la llamada, tal como se observa en la **Figura 3.6.**

Figura 3.6

Flujo de atención de mensajes entrantes



Una vez activado el flujo, el mensaje se interpreta y se aplica un control de entrada para manejar secuencias cercanas de mensajes.

A continuación, el flujo enruta la entrada según el tipo de contenido; si existe voz, se aplica un subproceso de preparación antes de invocar al agente.

Finalmente, el resultado se envía de regreso al canal como texto. Con este diseño, el canal WhatsApp queda separado del procesamiento de IA, y la cola actúa como amortiguador, lo cual mejora la estabilidad operativa cuando existen picos de mensajes o variaciones del canal.

- **Visualización operativa en pantalla**

La visualización operativa se implementa mediante una Raspberry Pi que accede, a través de un navegador web, a una sección específica de la interfaz orientada a la visualización de las citas del día. Este dispositivo actúa únicamente como un punto de visualización, sin incorporar lógica de negocio adicional.

La información mostrada es provista por el mismo backend que alimenta la interfaz principal, garantizando consistencia en los datos visualizados. Esta pantalla permite al personal de la clínica monitorear el flujo de atención de manera continua, sin necesidad de interacción directa con el sistema.

3.1.3 Integración y despliegue

La integración extrema a extremo se consolidó mediante dos tipos de disparadores: eventos iniciados por acciones del personal de la clínica en la interfaz web y eventos iniciados por mensajes entrantes del canal WhatsApp.

En el primer caso, cuando el personal de la clínica registra citas o administra documentos, el backend consolida los cambios en el esquema del workspace correspondiente y, cuando se trata de documentos para el asistente conversacional, activa los webhooks que alimentan los flujos de trabajos de n8n.

En el segundo caso, cuando un cliente escribe por WhatsApp, Evolution API publica el mensaje en RabbitMQ y el workflow de n8n se encarga de procesarlo, invocar al agente con recuperación en Qdrant cuando aplica y devolver la respuesta al canal.

Esta integración asegura que la operación interna y el canal externo compartan una misma base de información y automatización, pero sin acoplar el procesamiento conversacional directamente al backend de atención de UI.

El despliegue se realizó mediante Portainer, donde se levantan los servicios requeridos por la solución, incluyendo el backend de la aplicación, el orquestador de automatización n8n, la base de datos vectorial Qdrant, el sistema de mensajería RabbitMQ y los servicios de soporte asociados a la persistencia y almacenamiento de documentos.

Esta forma de despliegue permite administrar los contenedores, reiniciar servicios y mantener el sistema operativo sin complejidad adicional para la clínica. Además, esta organización deja preparado el escenario para monitoreo y mejora iterativa, ya que los puntos principales de observación quedan claramente definidos: el backend para operaciones de negocio, n8n para ejecución de flujos de trabajos, RabbitMQ para tráfico de mensajes entrantes y Qdrant para el estado del índice semántico.

Capítulo 4

4.1 Análisis e interpretación de resultados

El presente capítulo expone el análisis e interpretación de los resultados obtenidos posteriormente a la implementación, integración y despliegue del sistema desarrollado, con el propósito de evaluar el cumplimiento de los objetivos específicos planteados en esta investigación.

La evaluación se llevó a cabo mediante sesiones de prueba controladas, orientadas a verificar el comportamiento del sistema en un contexto cercano al uso real. El análisis se apoya en los indicadores definidos en el Capítulo 2, los cuales permiten evaluar la automatización de la gestión de citas, la atención de mensajes sin intervención humana, el tiempo de respuesta del sistema ante mensajes entrantes y la actualización de la base de conocimiento utilizada por el asistente conversacional.

Se realizaron al menos tres sesiones de prueba; sin embargo, para efectos del presente análisis se consideran principalmente los resultados obtenidos en la última sesión, al corresponder a la versión más estable y completa del sistema. Los resultados se interpretan a partir de los indicadores establecidos, complementados con observaciones cualitativas obtenidas durante la interacción del personal de la clínica con la plataforma y el asistente conversacional.

4.2 Métricas asociadas al Objetivo Específico 1

4.2.1 Integración de módulos de inteligencia artificial

El primer objetivo se basa en integrar módulos de inteligencia artificial para la automatización de la gestión de citas. El primer objetivo específico plantea la integración de módulos de inteligencia artificial que permitan automatizar la gestión de citas y el envío de notificaciones automáticas, con el fin de optimizar la atención al cliente y alcanzar un nivel de automatización superior al 80% en los procesos programados. Para este propósito, el sistema incorpora un asistente conversacional a través del canal WhatsApp y una plataforma web utilizada por el personal de la clínica para la gestión interna de las citas.

4.2.2 Metodología de evaluación

Durante las sesiones de prueba se ejecutaron solicitudes reales asociadas a la gestión de citas, incluyendo agendamiento, reprogramación, cancelación y atención de casos fuera de alcance mediante derivación al personal de la clínica.

El indicador principal considerado fue ***el porcentaje de automatización de la gestión de citas***, definido como la proporción de solicitudes relacionadas con citas que fueron resueltas automáticamente por el sistema, sin intervención humana directa, respecto al total de solicitudes evaluadas durante la sesión.

Se consideró como solicitud resuelta automáticamente aquella en la que el asistente conversacional solicitó la información necesaria, confirmó la acción y registró correctamente la cita en la plataforma.

De manera complementaria, se analizó el tiempo aproximado de respuesta del sistema, entendido como el intervalo entre el mensaje enviado por el tutor de la mascota y la respuesta generada por el asistente conversacional. Este tiempo fue estimado mediante la observación directa del intercambio de mensajes y el registro de ejecuciones del flujo de automatización en n8n.

4.2.3 Resultados obtenidos

Tabla 4.1

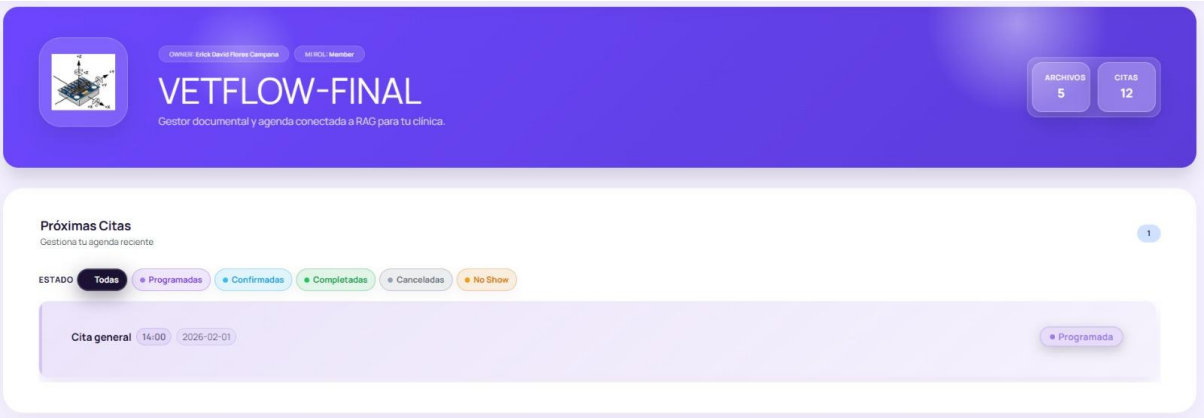
Resultados de solicitudes de gestión de citas

Solicitud	Tipo	Resultado	Intervención humana
S1	Agendamiento	Exitosa	No
S2	Reprogramación	Exitosa	No
S3	Cancelación	Exitosa	No
S4	Agendamiento	Exitosa	No
S5	Agendamiento	Exitosa	No
S6	Reprogramación	Exitosa	No
S7	Cancelación	Exitosa	No
S8	Agendamiento	Exitosa	No
S9	Emergencia Simulada	Derivación	Sí

Durante la sesión de prueba analizada se registraron un total de nueve solicitudes relacionadas con la gestión de citas a través del canal WhatsApp. Como se observa en la **tabla 4.1,** de un total de 9 solicitudes evaluadas, 8 fueron resueltas automáticamente por el sistema y 1 requirió derivación al personal de la clínica.

Figura 4.1

Visualización de citas registradas en la interfaz web de la plataforma.



Aplicando la fórmula definida en el Capítulo 2, el porcentaje de automatización obtenido fue 88,9%.

$$Automatización = \left(\frac{8}{9}\right) \times 100 = 88.9\% \quad (4.1)$$

El resultado de la ecuación 4.1 se respalda visualmente mediante la evidencia de citas correctamente registradas en la interfaz web, como se observa en la **Figura 4.1**, así como en la confirmación automática enviada al tutor de la mascota, mostrada en la **Figura 4.2**.

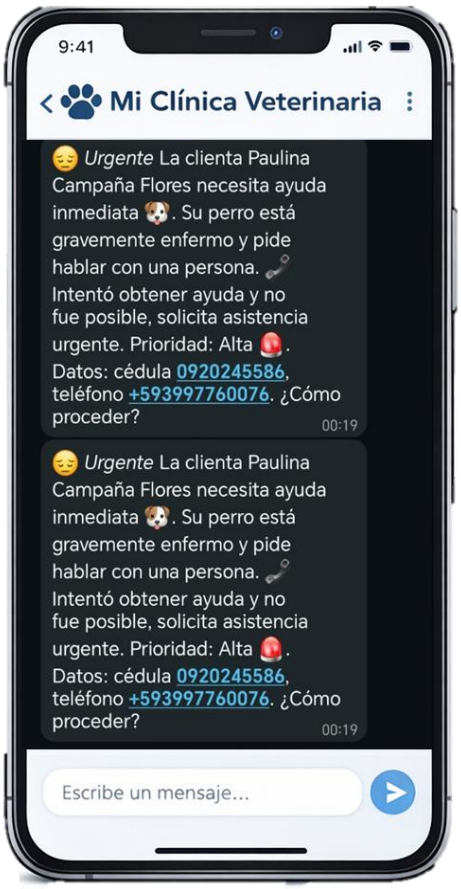
Figura 4.2

Confirmación automática de cita generada por el asistente conversacional a través de WhatsApp.

En los casos fuera de alcance, el sistema generó mensajes de seguridad y realizó la derivación correspondiente al personal de la clínica. Un ejemplo de este comportamiento se presenta en la **Figura 4.3**, donde se evidencia el mensaje de derivación emitido por el asistente conversacional.

Figura 4.3

Mensaje de derivación al personal de la clínica ante un caso fuera de alcance.



El tiempo de ejecución del flujo de automatización fue medido a partir de los registros internos del sistema en el panel de ejecuciones de n8n. Para esta medición se consideraron únicamente ejecuciones exitosas del flujo de automatización asociadas al procesamiento automático de mensajes, excluyendo aquellas que permanecen activas en nodos de espera (buffer), ya que estas dependen del tiempo de interacción del usuario y no del procesamiento interno del sistema.

Tabla 4.2

Tiempo de ejecución del flujo de automatización registrado en n8n

Ejecución	Tiempo registrado
1	37 ms
2	23 ms
3	26 ms
4	54 ms

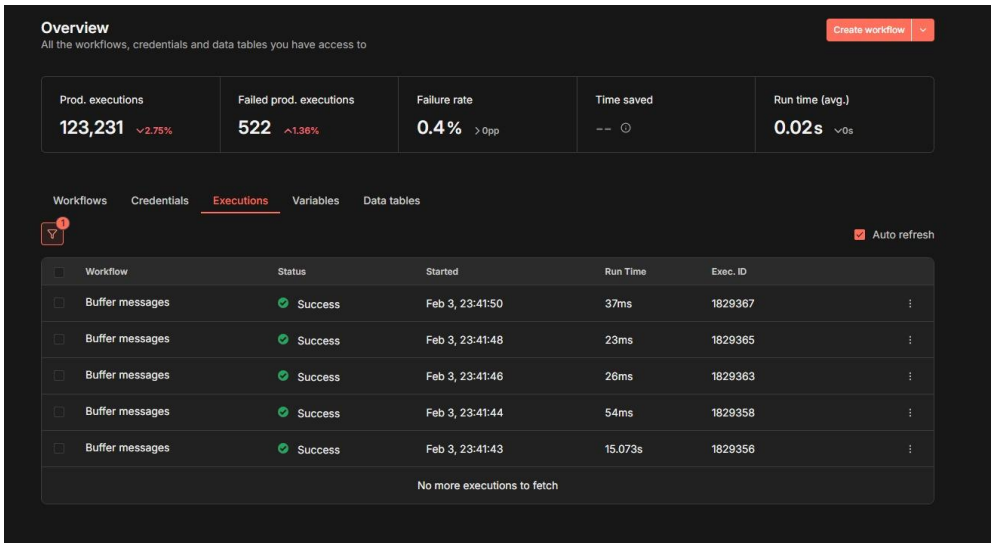
Tomando los datos de la **tabla 4.2,** el tiempo promedio de ejecución del flujo fue de:

$$Tiempo\ promedio\ = \frac{(0.037+0.023+0.026+0.054)}{4} = 0.035\ segundos \qquad (4.2)$$

El resultado de la ecuación 4.2 evidencia que el procesamiento automático del sistema se realiza en milisegundos, por lo que la latencia percibida por el usuario depende principalmente del tiempo de interacción humana y no del desempeño del motor de automatización.

Figura 4.4

Ejecuciones exitosas del flujo de automatización de gestión de citas en n8n.



La ejecución automática de estos flujos se evidencia en el panel de ejecuciones de n8n, donde se observan múltiples ejecuciones exitosas del flujo de procesamiento de mensajes, tal como se presenta en la **Figura 4.4**.

4.2.4 Análisis e interpretación

Los resultados cuantitativos obtenidos permiten analizar el cumplimiento del Objetivo Específico 1 desde dos dimensiones: el nivel de automatización alcanzado y el desempeño del procesamiento interno del sistema.

En primer lugar, ***el porcentaje de automatización*** obtenido fue del 88,9%, calculado a partir de 8 solicitudes resueltas automáticamente de un total de 9 evaluadas durante la sesión de prueba. Este valor supera el umbral del 80% establecido como meta en el objetivo específico, lo que indica que la mayoría de las solicitudes relacionadas con la gestión de citas pueden ser atendidas sin intervención humana directa.

No obstante, se registró un caso que requirió derivación al personal de la clínica. El análisis de esta situación evidenció que la limitación no estuvo asociada a fallos técnicos en la automatización, sino a aspectos relacionados con la interpretación contextual del mensaje y el diseño de instrucciones (*prompt engineering*) del asistente conversacional.

Este resultado pone de manifiesto la necesidad de refinar los parámetros del modelo para mejorar el manejo de escenarios ambiguos o fuera del flujo estándar de citas.

En segundo lugar, el tiempo promedio de ejecución del flujo de automatización registrado en n8n fue de 0,035 segundos por ejecución, lo cual demuestra que el procesamiento interno del sistema se realiza en milisegundos. Esto confirma que la latencia percibida en la interacción no está asociada al motor de automatización, sino principalmente al tiempo de respuesta del usuario durante la conversación.

En conjunto, los resultados evidencian que la integración de módulos de inteligencia artificial permitió automatizar de manera efectiva la gestión de citas, alcanzando un nivel de automatización superior al objetivo planteado. Sin embargo, también se identifican oportunidades de mejora en el ajuste del asistente conversacional para reducir aún más los casos que requieren intervención humana.

4.3 Métricas asociadas al Objetivo Específico 2

4.3.1 Construir una base de conocimiento dinámica e interactiva

El segundo objetivo específico se orienta a la construcción de una base de conocimiento dinámica que permita al sistema proporcionar respuestas contextualizadas a partir de información administrada por el personal de la clínica. De acuerdo con lo definido en el Capítulo 2, el indicador asociado a este objetivo es la **frecuencia de actualización de la base de conocimiento**, entendida como la capacidad del sistema para incorporar nuevos documentos y reflejar dicha información en las respuestas generadas por el asistente conversacional.

4.3.2 Metodología de evaluación

La evaluación de este objetivo se realizó en el marco de las pruebas de extremo a extremo descritas en la Fase 5 del Capítulo 2. Durante las sesiones de prueba, el personal de la clínica cargó documentos informativos en la plataforma, los cuales fueron procesados e incorporados a la base de conocimiento del sistema. Posteriormente, se realizaron consultas relacionadas con el contenido de dichos documentos a través del asistente conversacional.

La validación se apoyó en dos elementos principales: (i) la observación de las actualizaciones realizadas en la base de conocimiento durante la sesión de prueba, y (ii) la evaluación cualitativa de las respuestas generadas por el asistente, utilizando una lista de chequeo de calidad del agente virtual aplicado durante la prueba (ver Apéndice A), el cual considera aspectos como detección de intención, coherencia de la respuesta, manejo de ambigüedad y tono del lenguaje.

4.3.3 Resultados obtenidos

Durante la sesión de prueba se registraron múltiples actualizaciones de la base de conocimiento mediante la carga de documentos por parte del personal de la clínica. Estas actualizaciones fueron procesadas por el sistema y utilizadas posteriormente como fuente de información para responder consultas realizadas por los tutores de las mascotas.

Figura 4.5

Respuesta de la asistente conversacional basada en información actualizada

En la **Figura 4.5** se presenta un ejemplo de una respuesta generada por el asistente conversacional a partir de información contenida en los documentos cargados, lo cual evidencia que la base de conocimiento es utilizada activamente durante la interacción.

Sin embargo, también se identificaron casos en los que el asistente generó respuestas ambiguas o imprecisas, principalmente cuando la información disponible no estaba estructurada de forma clara o cuando la consulta realizada presentaba ambigüedad contextual. Como medida de mejora, se identificó la necesidad de ajustar las instrucciones dadas al asistente conversacional, incorporando instrucciones más específicas para evitar respuestas basadas en supuestos y priorizar la solicitud de aclaraciones ante consultas ambiguas. Asimismo, se recomendó estructurar los documentos cargados con información más segmentada y explícita para reducir ambigüedad en la recuperación semántica.

Figura 4.6

Ejemplo de respuesta ambigua generada por el asistente conversacional

Un ejemplo de este comportamiento se muestra en la **Figura 4.6**, donde el asistente genera una respuesta basada en supuestos ante una consulta ambigua. Estas situaciones fueron registradas como parte de las observaciones cualitativas del personal de la clínica.

4.3.4 Análisis e interpretación

Los resultados obtenidos evidencian que el sistema permite la incorporación de nuevos documentos y su utilización inmediata en la generación de respuestas, confirmando el carácter dinámico de la base de conocimiento. Sin embargo, la calidad de las respuestas generadas depende directamente de la claridad y estructura de la información cargada por la clínica, así como de la configuración de las instrucciones del asistente conversacional.

En los casos en que los documentos presentaban información ambigua o poco estructurada, se observaron respuestas menos precisas, lo que indica que la confiabilidad del sistema no depende únicamente del modelo de inteligencia artificial, sino también de la calidad de los datos de entrada y del diseño de las instrucciones del agente.

Por tanto, el sistema cumple el objetivo de mantener una base de conocimiento dinámica desde el punto de vista funcional. No obstante, la confiabilidad observada durante la validación fue adecuada en la mayoría de los casos evaluados, aunque condicionada por la calidad del contenido cargado y por la configuración de las instrucciones del asistente, lo que evidencia la necesidad de ajustes para fortalecer la consistencia de las respuestas.

4.4 Métricas asociadas al Objetivo Específico 3

4.4.1 Desarrollar un módulo de visualización en tiempo real

El tercer objetivo específico plantea el desarrollo de un módulo de visualización en tiempo real que permita al personal de la clínica acceder a información relevante sobre las citas, garantizando una latencia de respuesta no mayor a 5 segundos. Este módulo busca apoyar la organización interna de la clínica mediante la visualización oportuna del estado de la agenda y sus actualizaciones.

4.4.2 Metodología de evaluación

La evaluación de este objetivo se realizó mediante una sesión práctica con el personal de la clínica, durante la cual se utilizó el módulo de visualización para revisar la agenda diaria, identificar citas según su estado y observar la actualización de la información tras la modificación de una cita.

La latencia del sistema se evaluó mediante **observación directa**, midiendo el tiempo transcurrido entre la acción realizada en la interfaz (recarga de la agenda, cambio de estado o reprogramación de una cita) y la visualización efectiva del cambio en pantalla. Para esta medición se utilizó un cronómetro, registrando tiempos aproximados en segundos y excluyendo el tiempo de interacción humana.

4.4.3 Resultados obtenidos

La **Figura 4.7** muestra el módulo de visualización utilizado durante la sesión de prueba. La evaluación se centró en medir el tiempo que tarda el sistema en reflejar cambios realizados en la agenda.

Figura 4.7

Módulo de visualización de la agenda diaria en la plataforma web.

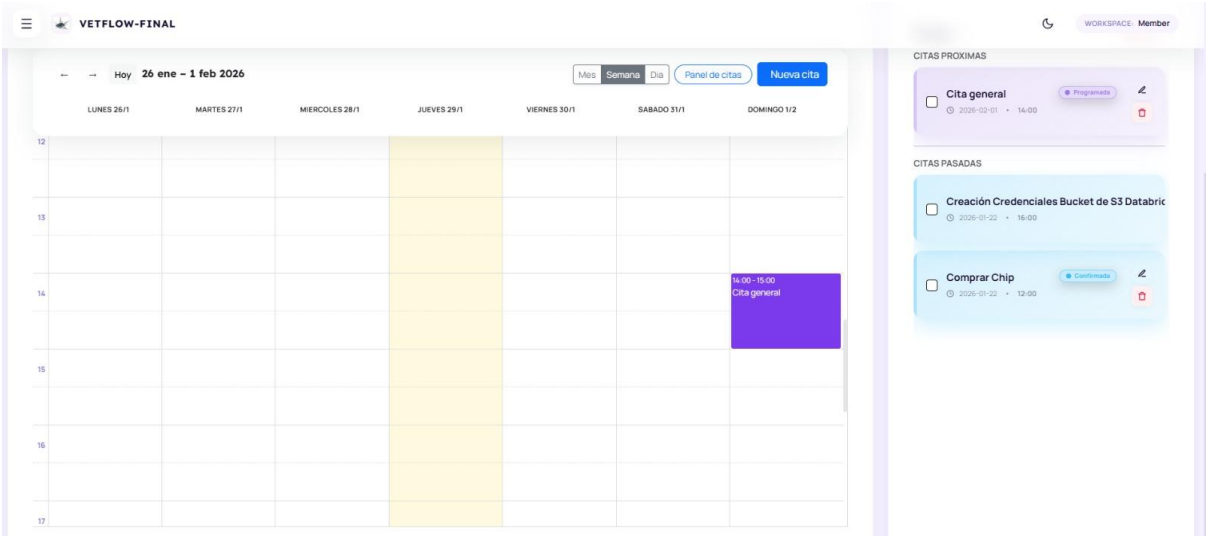


Tabla 4.3

Latencia medida del módulo de visualización

Acción evaluada	Medición 1 (s)	Medición 2 (s)	Medición 3 (s)	Promedio (s)
Carga de agenda	2.3	2.1	2.4	2.27
Cambio de estado	2.1	2.8	3.0	3.00
Reprogramación	2.5	2.8	2.4	2.50

Nota. Mediciones realizadas mediante cronómetro durante la sesión de prueba.

Los tiempos promedio obtenidos para las acciones evaluadas se presentan en la **Tabla 4.3**. En todos los casos, la latencia promedio registrada fue inferior a los 5 segundos establecidos como criterio, cumpliendo así el requisito definido en el objetivo específico.

Figura 4.8

Actualización de la información de una cita tras el cambio de estado

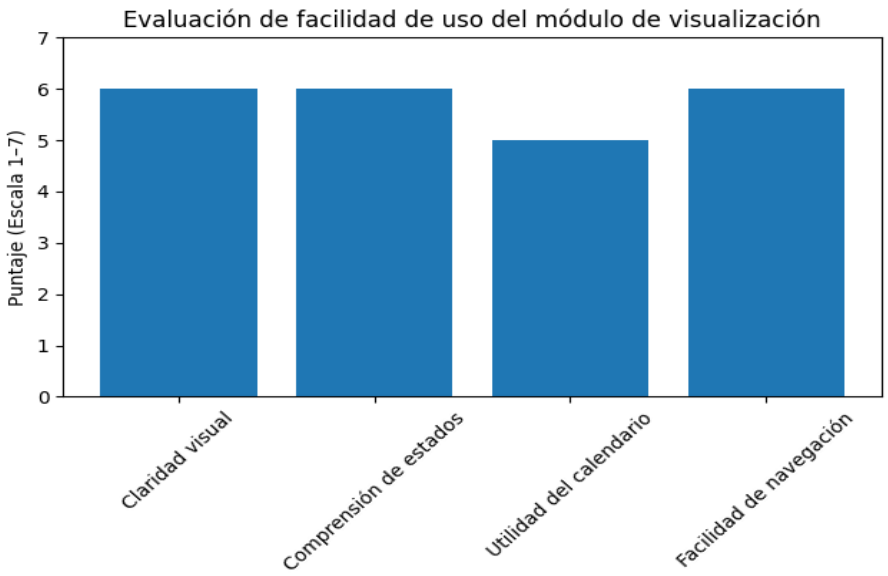
Adicionalmente, tal como se ve en la **Figura 4.9** el módulo fue visualizado desde una Raspberry Pi conectada a una pantalla, accediendo a la misma interfaz web evaluada durante la sesión. Esto permitió confirmar que la información presentada en el punto físico de visualización corresponde al estado actualizado de la agenda, tal como se muestra en la **Figura 4.8**.

Figura 4.9

Visualización operativa del módulo desde Raspberry Pi.

Figura 4.10

Evaluación de facilidad de uso del módulo de visualización



En cuanto a la experiencia de uso, se aplicó una escala de facilidad (1–7) incluida en el instrumento de evaluación. La **Figura 4.10** muestra como los puntajes registrados para los ítems relacionados con claridad visual, comprensión de estados y utilidad del calendario se ubicaron en valores superiores a 5 sobre 7, indicando una percepción favorable del módulo por parte del personal de la clínica.

4.4.4 Análisis e interpretación

Los resultados cuantitativos de latencia demuestran que el módulo de visualización cumple el requisito técnico definido, ya que el tiempo promedio de actualización en todas las acciones evaluadas se mantiene por debajo del umbral de 5 segundos. Esto evidencia que el sistema responde de manera oportuna ante modificaciones realizadas en la agenda.

Desde la perspectiva de experiencia de usuario, las valoraciones obtenidas en la escala aplicada reflejan una percepción positiva en cuanto a claridad y utilidad del módulo. Las observaciones registradas durante la sesión se concentraron en aspectos de organización visual, los cuales representan oportunidades de mejora en el diseño, pero no afectan el desempeño técnico ni el cumplimiento del objetivo.

En consecuencia, los datos obtenidos permiten afirmar, con base en mediciones registradas, que el módulo desarrollado cumple con el objetivo específico de visualización en tiempo real, tanto desde el punto de vista técnico (latencia) como desde la percepción de uso en un entorno operativo controlado.

Capítulo 5

5.1 Conclusiones y recomendaciones

El presente capítulo sintetiza las conclusiones derivadas del desarrollo e implementación de la plataforma integral de gestión y soporte al cliente para clínicas veterinarias, así como las recomendaciones y líneas futuras de trabajo, establecidas a partir de los resultados obtenidos en las sesiones de prueba del sistema. La evaluación permitió verificar el desempeño funcional del bot de WhatsApp y de la plataforma web, considerando métricas asociadas a automatización, precisión de respuestas y visualización en tiempo real.

5.2 Conclusiones

El presente trabajo tuvo como objetivo el diseño, implementación y validación de una plataforma integral de gestión y soporte al cliente para clínicas veterinarias, apoyada en automatización e inteligencia artificial conversacional. A partir de los resultados obtenidos durante las sesiones de prueba y el análisis desarrollado en el Capítulo 4, se concluye que el sistema cumple de manera satisfactoria con los objetivos específicos planteados.

En relación con el **Objetivo Específico 1**, orientado a la automatización de la gestión de citas y notificaciones, los resultados evidencian que el sistema alcanzó un nivel de automatización del 88,9% en las solicitudes evaluadas durante la sesión de validación, superando el umbral del 80% definido como criterio. La mayoría de las solicitudes de agendamiento, reprogramación y cancelación de citas fueron resueltas de forma automática a través del asistente conversacional, sin intervención directa del personal de la clínica. El porcentaje restante correspondió a un caso fuera de alcance que requirió derivación manual, lo cual se encontraba contemplado dentro del diseño del sistema. Esto permitió reducir la carga operativa asociada a tareas repetitivas y mejorar la atención al cliente, confirmando la efectividad de la integración de módulos de inteligencia artificial en los flujos operativos.

Respecto al **Objetivo Específico 2**, enfocado en la construcción de una base de conocimiento dinámica e interactiva, los resultados obtenidos durante la validación muestran que el sistema permite la incorporación de nuevos documentos y su utilización inmediata en las respuestas generadas. Durante las pruebas se registraron múltiples actualizaciones de la base de conocimiento, las cuales fueron reflejadas en las respuestas del sistema, demostrando

su carácter dinámico. No obstante, también se identificaron limitaciones en el manejo de consultas ambiguas y en la coherencia de algunas respuestas, principalmente asociadas a la estructura y calidad de la información cargada. Estas observaciones indican que, si bien el objetivo se cumple a nivel funcional, existen oportunidades claras de mejora para fortalecer la fiabilidad del sistema.

En cuanto al **Objetivo Específico 3**, relacionado con el desarrollo de un módulo de visualización en tiempo real, los tiempos promedio registrados para las acciones evaluadas fueron inferiores a 5 segundos, cumpliendo el criterio de latencia establecido en el objetivo específico. Las mediciones realizadas mediante observación directa confirmaron que las acciones evaluadas se reflejan en pantalla de forma oportuna, tanto en la interfaz web como en el punto físico de visualización implementado mediante una Raspberry Pi. Este módulo facilita la organización interna de la clínica y proporciona una visión clara del estado de la agenda en tiempo real.

De manera global, se concluye que la solución desarrollada integra de forma efectiva automatización, inteligencia artificial conversacional y visualización en tiempo real, constituyéndose como una herramienta viable para apoyar la gestión operativa y la atención al cliente en clínicas veterinarias. La arquitectura implementada permitió desacoplar responsabilidades y facilitar la validación funcional del sistema en un entorno cercano al uso real.

5.3 Líneas futuras de investigación y mejora

A partir de los resultados obtenidos y de las observaciones realizadas durante la validación del sistema, se identifican diversas líneas de trabajo que pueden ser abordadas en investigaciones futuras o como mejoras evolutivas de la plataforma.

Una primera línea de mejora consiste en fortalecer los mecanismos de validación de las respuestas del asistente conversacional, incorporando estrategias adicionales para el manejo de ambigüedad, tales como preguntas cerradas, confirmaciones explícitas o restricciones contextuales, así como ajustes en el diseño de instrucciones dentro del asistente conversacional. Esto permitiría aumentar la fiabilidad de las respuestas y mejorar la experiencia del usuario en consultas complejas.

Otra línea relevante es la optimización de la base de conocimiento, mediante la estandarización de los documentos cargados y la incorporación de metadatos estructurados. Esto podría contribuir a reducir respuestas imprecisas y facilitar una recuperación de información más consistente durante las interacciones conversacionales.

Asimismo, se plantea como trabajo futuro la evaluación del sistema en un entorno de producción por un periodo prolongado, lo cual permitiría medir indicadores de desempeño a largo plazo, como disponibilidad, estabilidad y patrones de uso reales. Este tipo de evaluación ampliaría el alcance de los resultados obtenidos en este estudio, que se centró en un periodo de validación controlado.

En el ámbito de la visualización, se identifican oportunidades de mejora relacionadas con el diseño de la interfaz, tales como la optimización del calendario, la diferenciación del horario laboral y la reducción de elementos visuales no relevantes. Estas mejoras podrían incrementar la usabilidad del sistema sin afectar su arquitectura ni su rendimiento técnico.

Finalmente, se considera como línea futura la integración con otros sistemas clínicos, como historiales médicos electrónicos o sistemas de facturación, así como la ampliación del soporte a nuevos canales de comunicación. Estas extensiones permitirían consolidar la plataforma como una solución integral para la gestión y atención al cliente en el contexto veterinario.

Referencias

About: Node-RED. (s. f.). Recuperado 18 de enero de 2026, de https://nodered.org/about/

Abqari, U., van ’t Noordende, A. T., Richardus, J. H., Isfandiari, M. A., & Korfage, I. J. (2022). Strategies to promote the use of online health applications for early detection and raising awareness of chronic diseases among members of the general public: A systematic literature review. International Journal of Medical Informatics, 162, 104737. https://doi.org/10.1016/j.ijmedinf.2022.104737

Aquino, S. (2024). What is a Vector Database? - Qdrant. Qdrant. https://qdrant.tech/articles/what-is-a-vector-database/

Arifah Fasha, R., Muhammad Harith, M., & Universiti Teknologi MARA Cawangan

Balto, R. (2025, octubre 7). El impacto de la transformación digital en una clínica veterinaria. *Revista Balto. Informativo para veterinarios especializados en animales de compañía*. https://revistabalto.com/2025/10/el-impacto-de-la-transformacion-digital-en-una-clinica-veterinaria/

Beyer, K. (2023). Customer experiences drive the need of innovation in veterinary practices. *Procedia Computer Science, 225*, 3094–3103. https://doi.org/10.1016/j.procs.2023.10.303

Bin, C. (2024, diciembre 16). Cómo instalar Portainer (Tutorial para desarrolladores). *Ironsoftware*. https://ironsoftware.com/es/enterprise/securedoc/blog/using-ironsecuredoc/how-to-install-portainer/

Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G.,

Henighan, T., Child, R., Ramesh, A., Ziegler, D. M., Wu, J., Winter, C., … Amodei, D. (2020). Language models are few-shot learners (No. arXiv:2005.14165). *arXiv*. https://doi.org/10.48550/arXiv.2005.14165

Cachuela, S. J., Matias, D. M., Jose, R. J., & Altura, J. (2025). Navigating veterinary practice in the digital age: Implementing a web-based information management system at Animals' Choice Clinic. *Psychology and Education: A Multidisciplinary Journal, 37*. https://doi.org/10.70838/pemj.370506

Chávez Yagual, D. S. (2022). Desarrollo de una aplicación web para el agendamiento de citas y control de procesos de la clínica veterinaria Animal Health. https://repositorio.upse.edu.ec/handle/46000/8674

Cole, S. (2024, julio 15). ¿Qué es la IA multimodal? IBM. https://www.ibm.com/es-es/think/topics/multimodal-ai

Cotrina Cabezas, W. G. (2022). Software como servicio con chatbot para el proceso de gestión de citas en la clínica veterinaria Mascotas Club. Repositorio Institucional - UCV. https://repositorio.ucv.edu.pe/handle/20.500.12692/151126

Covetrus. (2025). How to achieve optimal reminder and recall effectiveness [Covetrus]. https://covetrus.com/insights/how-to-achieve-optimal-reminder-and-recall-effectiveness/

Definition of Hyperautomation—Gartner Information Technology Glossary. (s. f.). Gartner. Recuperado 25 de noviembre de 2025, de https://www.gartner.com/en/information-technology/glossary/hyperautomation

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2018, octubre 11). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. arXiv.Org. https://arxiv.org/abs/1810.04805v2

Dmitrievich, R., & Alexandrovich, V. (2024). THE USE OF DIGITAL TECHNOLOGIES IN VETERINARY MEDICINE AND VETERINARY RECORDS MANAGEMENT: ANALYSIS OF THE PRIVATE AND PUBLIC SECTORS. ResearchGate. https://doi.org/10.35679/1991-9476-2024-19-2-352-359

EvolutionAPI. (2023). *WhatsApp Business API*. https://www.evolutionapi.com/

Fejzic, N., Kraišnik, I., & Bajić, A. (2024). Digital Solutions for Veterinary Organizations: Solutions for Growing Client Expectations and Optimization of Time, Resources, Knowledge, and Skills of Veterinarians. ResearchGate. https://www.researchgate.net/publication/386328807_Digital_Solutions_for_Veterinary_Organizations_Solutions_for_Growing_Client_Expectations_and_Optimization_of_Time_Resources_Knowledge_and_Skills_of_Veterinarians

Fernández, M. L. (2025, agosto 21). Modelos multimodales en IA: Cuando texto, imagen y audio convergen. Modelos multimodales: cuando texto, imagen y audio convergen. https://www.automatizapro.com.ar/blog/modelos-multimodales-texto-imagen-audio/

Gadiraju, S. S., Liao, D., Kudupudi, A., Kasula, S., & Chalasani, C. (2024). InfoTech Assistant: A Multimodal Conversational Agent for InfoTechnology Web Portal Queries. 2024 IEEE International Conference on Big Data (BigData), 3264-3272. https://doi.org/10.1109/BigData62323.2024.10825668

Gregorio, N. H. A., & Angel, S. T. J. (2021). INGENIERO EN COMPUTACIÓN E INFORMÁTICA.

Huong, P. T., Hien, L. T., Son, N. M., Tuan, H. C., & Nguyen, T. Q. (2025). Enhancing deep convolutional neural network models for orange quality classification using MobileNetV2 and data augmentation techniques. Journal of Algorithms & Computational Technology, 19, 17483026241309070. https://doi.org/10.1177/17483026241309070

ISO/IEC 25002:2024(en), Systems and software engineering—Systems and software Quality Requirements and Evaluation (SQuaRE)—Quality model overview and usage. (s. f.). Recuperado 19 de enero de 2026, de https://www.iso.org/obp/ui/en/#iso:std:78175:en

ISO/IEC 25023:2016(en), Systems and software engineering—Systems and software Quality Requirements and Evaluation (SQuaRE)—Measurement of system and software product quality. (s. f.). Recuperado 19 de enero de 2026, de https://www.iso.org/obp/ui/en/#iso:std:iso-iec:25023:ed-1:v1:en

Kammrath Betancor, P., Boehringer, D., Jordan, J., Lüchtenberg, C., Lambeck, M., Ketterer, M. C., Reinhard, T., & Reich, M. (2025). Efficient patient care in the digital age: Impact of online appointment scheduling in a medical practice and a university hospital on the "no-show"-rate. Frontiers in Digital Health, 7. https://doi.org/10.3389/fdgth.2025.1567397

Kulkarni, M., Tangarajan, P., Kim, K., & Trivedi, A. (2024). Reinforcement Learning for Optimizing RAG for Domain Chatbots (No. arXiv:2401.06800). arXiv. https://doi.org/10.48550/arXiv.2401.06800

Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., Küttler, H., Lewis, M., Yih, W., Rocktäschel, T., Riedel, S., & Kiela, D. (2020). Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks. Advances in Neural Information

Processing Systems, 33, 9459-9474. https://proceedings.neurips.cc/paper/2020/hash/6b493230205f780e1bc26945df7481e5-Abstract.html

Li, Z., Wang, Z., Wang, W., Hung, K., Xie, H., & Wang, F. L. (2025). Retrieval-augmented generation for educational application: A systematic survey. Computers and Education: Artificial Intelligence, 8, 100417. https://doi.org/10.1016/j.caeai.2025.100417

Loor García, Y. Y. (2019). Desarrollo de aplicación web para la gestión de consultas y agendamiento de citas de mascota de la clínica veterinaria burgos. [bachelorThesis]. http://dspace.ups.edu.ec/handle/123456789/16991

Make. (s. f.). Make | AI Workflow Automation Software & Tools. Make. Recuperado 18 de enero de 2026, de https://www.make.com/en

Mell, P., & Grance, T. (2011). *The NIST definition of cloud computing*. NIST. https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-145.pdf

Microsoft. (2026a). *Azure OpenAI Service*. https://azure.microsoft.com/es-mx/solutions/ai

Microsoft. (2026b). *Design multi-tenant SaaS applications*. https://learn.microsoft.com/en-us/azure/architecture/guide/saas-multitenant-solution-architecture/

Microsoft. (2023c). *What is Azure Blob Storage?* https://learn.microsoft.com/en-us/azure/storage/blobs/storage-blobs-introduction

Pivotal. (2023). *RabbitMQ Documentation*. https://www.rabbitmq.com/

Twilio. (2023). *What is a Webhook?* https://www.twilio.com/docs/glossary/what-is-a-webhook

Umar, A. (2021). *End-to-End Testing: Definition, Process, and Best Practices*. BrowserStack. https://www.browserstack.com/guide/end-to-end-testing

Merritt, R. (2025, enero 31). What Is Retrieval-Augmented Generation aka RAG? NVIDIA Blog. https://blogs.nvidia.com/blog/what-is-retrieval-augmented-generation/

Modyeievsky, S. (2025, junio 20). Qué es lo que más preocupa a los veterinarios propietarios de su clínica veterinaria en la actualidad -4a entrega. Selecciones Veterinarias. https://www.seleccionesveterinarias.com/que-es-lo-que-mas-preocupa-a-los-veterinarios-propietarios-de-su-clinica-veterinaria-en-la-actualidad-4a-entrega/

Molloy, T. (2025). How to Optimize Flujos de trabajos in Veterinary Clinics | Improve Efficiency. VetPort. https://www.vetport.com/how-to-optimize-flujos de trabajos-in-veterinary-clinics

N8n Hosting Documentation and Guides | n8n Docs. (s. f.). Recuperado 18 de enero de 2026, de https://docs.n8n.io/hosting/

N8n-io/n8n. (2026). [TypeScript]. n8n - Workflow Automation. https://github.com/n8n-io/n8n (Obra original publicada en 2019)

Nebot, J. M. (2025, mayo 6). Retrieval Augmented Generation (RAG): ¿Qué debo saber para empezar? Telefónica Tech. https://telefonicatech.com/blog/retrieval-augmented-generation-rag-que-debo-saber-para-empezar

Ochoa, A. C., Murillo, A. C., & Rodas-Silva, J. (2020). El uso de aplicaciones Web para la Gestión de clínicas veterinarias y su incidencia en la mejora de procesos administrativos. Ecuadorian Science Journal, 5(Esp.4), 109-120.

OpenAI. (2024, enero 12). GPT-4. https://openai.com/es-ES/index/gpt-4-research/

Oracle. (2020, octubre 7). ¿Qué es un chatbot? https://www.oracle.com/es/chatbots/what-is-a-chatbot/

Oracle. (2021, noviembre 9). ¿Qué es Docker? | Oracle América Latina. https://www.oracle.com/latam/cloud/cloud-native/container-registry/what-is-docker/

Oriigin. (2025, octubre 13). Automatización con N8N: cómo conectar tus apps y crear flujos de trabajo sin esfuerzo. https://oriigin.co/

Perlis. (2023). An online scheduling platform for veterinary appointments. Jurnal Intelek, 18(2). https://doi.org/10.24191/ji.v18i2.22092

Q2BStudio. (2025, septiembre 29). Memoria de Contexto en Chatbots de IA: Por qué Importan los Mensajes de Ayer. https://www.q2bstudio.com/nuestro-blog/30134/memoria-de-contexto-en-chatbots-de-ia-por-que-importan-los-mensajes-de-ayer

Qdrant. (2023). https://qdrant.tech/documentation/

Schedule DAGs in Apache Airflow® | Astronomer Docs. (s. f.). Recuperado 18 de enero de 2026, de https://www.astronomer.io/docs/learn/scheduling-in-airflow

Stryker, C., & Kavlakoglu, E. (s. f.). What Is Artificial Intelligence (AI)? | IBM. Recuperado 25 de noviembre de 2025, de https://www.ibm.com/think/topics/artificial-intelligence

Sustainable Use License | n8n Docs. (s. f.). Recuperado 18 de enero de 2026, de https://docs.n8n.io/sustainable-use-license/

SYDLE. (2021, mayo 21). How to Perform AS IS, TO BE, and TO DO Process Mapping. Blog SYDLE. https://www.sydle.com/blog/as-is-to-be-to-do-process-mapping-60a81ebd22559e108ed7f51e

Towards an approach for developing and testing Node-RED IoT systems | Proceedings of the 1st ACM SIGSOFT International Workshop on Ensemble-Based Software Engineering. (s. f.). Recuperado 18 de enero de 2026, de https://dl.acm.org/doi/10.1145/3281022.3281023

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., & Polosukhin, I. (2023). Attention Is All You Need (No. arXiv:1706.03762). arXiv. https://doi.org/10.48550/arXiv.1706.03762

Vela Velastegui, J. F. (2020). Plan de negocios para la elaboración de una aplicación móvil para acceder a servicios veterinarios en el Distrito Metropolitano de Quito [bachelorThesis, Quito: Universidad de las Américas, 2020]. http://dspace.udla.edu.ec/handle/33000/13133

VetSoftwareHub. (2025, noviembre 22). Veterinary Scheduling Software That Reduces No-Shows & Phone Load. VetSoftwareHub. https://www.vetsoftwarehub.com/article/veterinary-scheduling-software-reduce-no-shows-phone-calls

Vetstoria. (2024, septiembre 27). Maximising Appointment Management at Your Veterinary Clinic. Vetstoria. https://www.vetstoria.com/blog/best-practices-scheduling-veterinary-clinic/

Weizenbaum, J. (1966). ELIZA—a computer program for the study of natural language communication between man and machine. Commun. ACM, 9(1), 36-45. https://doi.org/10.1145/365153.365168

What is Airflow®? —Airflow 3.1.5 Documentation. (s. f.). Recuperado 18 de enero de 2026, de https://airflow.apache.org/docs/apache-airflow/stable/index.html

What Is Hyperautomation? | IBM. (s. f.). Recuperado 25 de noviembre de 2025, de https://www.ibm.com/think/topics/hyperautomation

Williams, O., & Olajide, F. (2022). Towards the Design of an Intelligent Automation Framework for Business Processes. 13-17. https://doi.org/10.1109/ICICT55905.2022.00010

Zapier: Automate AI Flujos de trabajos, Agents, and Apps. (s. f.). Recuperado 18 de enero de 2026, de https://zapier.com/

Apéndice A

Instrumento de evaluación aplicado durante la sesión de validación del sistema
Proyecto: Plataforma integral de gestión y soporte al cliente para clínicas veterinarias
Fecha de aplicación: _____
Participante: Personal de la clínica
Duración aproximada: _____

1. Validación funcional – Gestión de citas (Canal WhatsApp)
Para cada escenario, marcar:
Éxito (1) / Fallo (0)

T1 – Agendar cita (cliente nuevo)
¿El asistente detectó correctamente la intención? (Sí / No)
¿Solicitó los datos necesarios? (Sí / No)
¿Confirmó antes de registrar la cita? (Sí / No)
¿La cita quedó registrada en la plataforma? (Sí / No)
Tiempo aproximado de resolución: _____
Resultado final: (1 / 0)
Observaciones:

_____

_____

T2 – Agendar cita (cliente existente)
¿Reconoció al cliente? (Sí / No)
¿Propuso horarios disponibles? (Sí / No)
¿Confirmó antes de registrar? (Sí / No)
¿La cita quedó registrada correctamente? (Sí / No)
Tiempo aproximado de resolución: _____
Resultado final: (1 / 0)
Observaciones:

_____

_____

T3 – Reprogramar cita
¿Identificó la cita correctamente? (Sí / No)
¿Permitió seleccionar nuevo horario? (Sí / No)
¿Confirmó el cambio? (Sí / No)
¿La modificación se reflejó en la agenda? (Sí / No)
Tiempo aproximado de resolución: _____
Resultado final: (1 / 0)
Observaciones:

_____

_____

T4 – Cancelar cita
¿Identificó la cita a cancelar? (Sí / No)

¿Solicitó confirmación? (Sí / No)
¿Actualizó el estado correctamente? (Sí / No)
Tiempo aproximado de resolución: _____
Resultado final: (1 / 0)
Observaciones:

_____

_____

T5 – Consulta frecuente
¿Respondió con información coherente? (Sí / No)
¿La respuesta fue clara y comprensible? (Sí / No)
¿Evita inventar información? (Sí / No)
Resultado final: (1 / 0)
Observaciones:

_____

_____

T6 – Caso fuera de alcance (emergencia)
¿Identificó que era un caso fuera de alcance? (Sí / No)
¿Emitió mensaje de seguridad adecuado? (Sí / No)
¿Derivó correctamente al personal? (Sí / No)
Resultado final: (1 / 0)
Observaciones:

_____

_____

2. Validación funcional – Plataforma web

T7 – Visualización de agenda
¿La agenda se carga correctamente? (Sí / No)
¿Los estados de citas son visibles? (Sí / No)
Tiempo de carga (segundos): _____
Resultado final: (1 / 0)
Observaciones:

_____

_____

T8 – Edición / Cambio de estado de cita
¿Permite modificar el estado? (Sí / No)
¿El cambio se refleja correctamente? (Sí / No)
Tiempo de actualización (segundos): _____
Resultado final: (1 / 0)
Observaciones:

_____

_____

T9 – Panel / Vista operativa
¿Se visualiza correctamente la información? (Sí / No)
¿Se actualiza tras modificaciones? (Sí / No)

Tiempo de actualización (segundos): _____
Resultado final: (1 / 0)
Observaciones:

_____

_____

3. Evaluación de facilidad de uso (Escala 1–7)

Escala:
1 = Muy difícil
7 = Muy fácil

Ítem evaluado	**Puntaje (1–7)**
Facilidad de navegación en la plataforma	_____
Claridad visual del calendario	_____
Comprensión de estados de cita	_____
Utilidad del módulo para la organización diaria	_____
Facilidad de interacción con el asistente	_____

4. Checklist de calidad del asistente conversacional

Marcar: Sí / No / Parcial

Criterio	**Evaluación**
Detecta correctamente la intención	_____
Solicita datos necesarios	_____
Confirma antes de ejecutar acciones	_____
Maneja adecuadamente ambigüedad	_____
Mantiene tono profesional	_____
Evita inventar información	_____
Realiza derivación cuando corresponde	_____
Registra correctamente las acciones	_____

5. Observaciones generales

_____

_____

_____

6. Propuestas de mejora

_____

_____

_____