



universidad
de león



GRADO EN INGENIERÍA INFORMÁTICA
Sistemas de Información y Gestión de Negocios

Asistente Inteligente para la Recomendación de Medicamentos Veterinarios

Autor: Jimena Onrubia Álvarez

Índice

[1.Introducción](#)

[2.Objeto de estudio y problema a resolver](#)

[3.Tecnologías y herramientas utilizadas](#)

[4.Resolución del problema y funcionamiento de la aplicación](#)

[1\) Fase de ingesta \(Scraping , csv y grafo\)](#)

[2\) Interacción con el usuario\(veterinario\)](#)

[3\) Motor de recomendación inteligente](#)

[4\) Generación de respuesta](#)

[5.Análisis DAFO de la aplicación](#)

[Debilidades](#)

[Amenazas](#)

[Fortalezas](#)

[Oportunidades](#)

[6.Conclusión](#)

[7.Bibliografía](#)

1.Introducción

En el contexto de la medicina veterinaria se exige el manejo de una gran cantidad de información farmacológica que se actualiza constantemente. Este proyecto se centra en el desarrollo de un sistema de recomendación inteligente de medicamentos para perros y gatos enfocado a asistir a veterinarios en la toma de decisiones.

Un LLM con una base de datos estructurada que permite al usuario (en este caso al veterinario) realizar consultas en lenguaje natural y obtener recomendaciones precisas o advertencias específicas según la especie, raza , enfermedad o síntomas.

Este sistema interpreta la sintomatología clínica y filtra los tratamientos aprobados legalmente en España.

2.Objeto de estudio y problema a resolver

El objeto de estudio es la optimización de la prescripción veterinaria mediante el uso de tecnologías NLP (Procesamiento de Lenguaje Natural).

El principal problema es la fragmentación y dificultad de acceso a la información oficial de medicamentos. Los veterinarios a menudo deben consultar prospectos a papel o bases de datos como CIMAVET que requieren búsquedas precisas y no permiten consultas por síntomas. Además el uso de IAs como ChatGPT u otras para medicina tiene el riesgo de “alucinaciones”, inventando dosis, medicamentos inexistentes o datos incorrectos para el uso de cualquier usuario final, no solo de veterinarios.

La solución propuesta resuelve estos problemas mediante el uso de una arquitectura híbrida RAG (Retrieval-Augmented Generation).No se deja al LLM “inventar” sino que utiliza un motor de recomendación determinista (smart_recommendation_engine.py) que busca datos reales (filtrando la consulta con Groq) y luego utiliza la IA (Groq/Llama-3) para la redacción y explicación de una respuesta profesional. Esto nos garantiza que la información es 100% verídica y basada en la Agencia Española de Medicamentos y Productos Sanitarios.

3. Tecnologías y herramientas utilizadas

Para el desarrollo del proyecto se ha utilizado un stack tecnológico basado en Python, priorizando eficiencia y capacidad de ejecución en tiempo real:

- Lenguaje : **Python 3.12**
- Interfaz de usuario : **Streamlit**. Elegido para crear la interfaz web de usuario profesional y funcional sin necesidad de un fronted complejo.
- Inteligencia Artificial/Modelo y razonamiento : **Groq API** modelo Llama-3.3-70b-Versatile.
 - Inicialmente se intentó usar con modelos locales Ollama como Llama 2, mistral o tinyllama pero se descartaron por problemas de memoria RAM y en otros casos falta de razonamiento lógico.
 - En este caso Groq se usa en dos enfoques diferentes dependiendo de la etapa.
 - Interpretación semántica (JSON): El modelo actúa como traductor estructurado convirtiendo el texto libre del veterinario en parámetros estandarizados JSON.
 - Generación de respuesta : Redacta el informe clínico final basándose exclusivamente en los datos recuperados, asegurando un tono profesional y técnico.
- Obtención de datos (Scraping) : **Selenium** y de forma manual exportando a csv de manera legal del catálogo público de CIMAVET.
- Procesamiento de datos : De los csvs y datos extraídos se utilizó Pandas y **JSON**. Se estructuró la información en un “Knowledge Graph” que mapea: SÍNTOMAS ↔ ENFERMEDADES ↔ MEDICAMENTOS ↔ DOSIS
- Entorno virtual : **venv** para gestión de dependencias aisladas.

4. Resolución del problema y funcionamiento de la aplicación

La aplicación funciona con un flujo de datos estructurado en 4 fases :

1) Fase de ingesta (Scraping , csv y grafo)

Previamente a la ejecución, el script de scrapping descarga los datos de CIMAVET y los que no era posible mediante scrapping se exportó manualmente a csv. Los datos obtenidos se procesan y cruzan entre ellos pasado a JSON que relacionan síntomas con patologías, dando lugar a un grafo de conocimiento local, robusto y fiable.

2) Interacción con el usuario(veterinario)

El veterinario accede a la web a través de un login cuyo nombre y contraseña solo la sabe personal autorizado para acceder a la consulta de medicamentos. Esta medida de seguridad se toma principalmente para evitar que usuarios que no sean veterinarios accedan a este tipo de información y hacer un mal uso de ella evitando que diagnostiquen o calculen dosis erróneas de medicamentos a sus mascotas. Después de este login de seguridad, el veterinario puede elegir dos métodos de consulta: Chat con IA y búsqueda en base de datos. En el caso de que sea el panel de chat, el veterinario introducirá el caso clínico en lenguaje natural.

3) Motor de recomendación inteligente

El núcleo del proyecto ha evolucionado a lo largo del proyecto de un sistema de reglas rígidas a un motor híbrido impulsado por IA. El flujo técnico es:

- A. Interpretación de Intención con Groq : Cuando el veterinario escribe una consulta, el motor envía el texto a Groq con instrucciones estrictas para devolver un objeto JSON estructurado. Esto nos permite normalizar términos automáticamente.
- B. Recuperación determinista (RAG) : Con el JSON limpio generado por IA, el motor busca en la base de datos local (enfermedades_loader.py y grafo de conocimiento).
Con esto conseguimos que la búsqueda en la base de datos o grafo sea 100% efectiva y de esta manera recupera las enfermedades y medicamentos exactos asociados a esa patología.

4) Generación de respuesta

Una vez tenemos los datos técnicos, se realiza una segunda llamada a la API de Groq.

- Prompt del sistema
- Contexto inyectado : Se le pasa el listado de medicamentos encontrados en la base de datos local / grafo de conocimiento.
- Resultado : La IA genera un informe clínico, a diferencia de un simple chat, utiliza terminología médica apropiada y presenta las opciones de tratamiento recuperadas de CIMAVET.

5. Análisis DAFO de la aplicación

Debilidades

Al realizar dos llamadas a Groq (la primera para interpretar JSON y la segunda para la respuesta), el tiempo de respuesta depende de la conexión a Internet.

Si el usuario no tiene conexión o el servidor de Groq cae, el sistema aunque tenga la base de datos local, no podría realizar consulta ni respuesta.

Fortalezas

- Comprensión semántica avanzada : Gracias al uso de Groq para interpretar la consulta, el sistema entiende cualquier variación lingüística compleja incluso errores tipográficos que un algoritmo tradicional fallaría.
- Normalización automática : Capacidad de traducir lenguaje coloquial de propietarios a términos médicos estandarizados para la búsqueda.
- Búsqueda rápida y precisa : El tiempo de respuesta del sistema es muy rápido y además hay un segundo panel donde el veterinario puede consultar la base de datos con los medicamentos (puede seleccionar perro , gato u ambos , si está autorizado o no y puede buscar directamente el medicamento o el principio activo) .

6.Conclusión

El proyecto es útil y puede ser un modelo base para otro tipo de sistemas de recomendación en el ámbito sanitario tanto para animales como personas aprovechando la lógica y estructura del sistema. También tiene un margen de mejora amplio donde se pueden añadir nuevas especies, medicamentos, síntomas o patologías.

Este proyecto o más bien esta asignatura me ha sido útil para aprender más a fondo sobre el funcionamiento y estructura interna de la IA (en este caso IAs especializadas en algún tema concreto), aprender sobre nuevas herramientas y cómo utilizarlas. Considero bastante útil para el futuro y sobre todo en nuestra carrera todos los temas hablados tanto en clase como investigados por nuestra propia cuenta al realizar el proyecto.

7.Bibliografía

- AEMPS. (2025). Centro de información Online de Medicamentos Veterinarios CIMAVET [:: CIMAVet :: Centro de información de medicamentos para veterinaria](#)
- Streamlit Inc. (2024). Streamlit Documentation [Streamlit documentation](#)
- Groq. (2025).GroqCloud API Documentation.[API Reference - GroqDocs](#)
- Selenium Project.(2024). Selenium WebDriver Documentation [WebDriver | Selenium](#)
- Youtube. videos sobre todas las herramientas utilizadas en el proyecto

