

TRABAJO DE GRADO

Opción Seminario

Asistente Virtual para Veterinarias: Chatbot Inteligente con Base de Datos Propia para Atención y Consultas

Reflexión sobre las estrategias metodológicas para la transformación digital en las prácticas contables aplicadas en el sector Veterinario

La transformación digital ha ido avanzando de manera rápida en todos los aspectos de la vida empresarial y el sector veterinario no es la excepción. Integrar soluciones tecnológicas aumenta la eficacia tanto como para el vendedor como para el cliente ayudando con esto también a optimizar los procesos internos.

A continuación, se destacan algunas estrategias tecnológicas implementado el uso de chat Bot inteligente con base de datos propia:

Automatización para la gestión contable: En cuanto a la automatización de procesos es muy importante ya que esta facilita de manera rápida consultar y agilizar dichos procesos, el asistente no solo ayuda a investigar estrategias para realizar de manera más fácil, sino que puede consultar en la base de datos propia dicha información para que sea más completa y precisa

Uso de base de datos propia para la gestión de información: El uso de la propia información en la base de datos ayuda a agilizar el proceso de recuperar información para ayudar al Chatbot a mostrar los datos de manera instantánea, sin necesidad de navegar por la web

Mejora la experiencia del cliente o del vendedor para tomar buenas decisiones: La toma de decisiones es muy importante a la hora de atender o ser atendido, nuestro asistente ayuda a consultar en la web para ayudar al vendedor a que le puede brindar a su cliente para el problema que tenga, como también el cliente puede encontrar información sobre cualquier problema relacionado con su mascota

Cumplimiento Normativo y Seguridad de la Información: El Chatbot está bien implementado con las normativas tomando derechos sobre la api realizada en este caso usamos 2 modelos siendo la más principal y más eficiente llama 3 por su rapidez, como también protegiendo los datos de la base de datos de manera local para que en caso de usarse por el vendedor solo este tenga acceso de lo contrario el Chatbot implementado para el cliente tendrá acceso a la información sobre el cuidado de su mascota esto con el fin de proteger los datos de la veterinaria.

Revisión sobre las estrategias metodológicas para la transformación digital en las prácticas contables aplicadas en el sector Veterinario

En el sistema realizado es muy importante destacar el uso de herramientas utilizados para realizar el Chatbot este impacta mucho hoy en día ya que es una herramienta que ayuda como una persona especializada para la toma de decisiones, haciendo que sea más rápida que una persona y mas exacta en cuanto a los problemas, utilizando en mi base de datos MySQL, api de Ollama y Python con sus librerías como también un servidor llamado Wamp que se utiliza para funcionar de manera local en la cual solamente el veterinario tendrá acceso a los datos esto para evitar que los clientes accedan de manera incorrecta a la información de la veterinaria

Automatización para la gestión contable: El chatbot es una herramienta clave en la toma de decisiones contables, ya que permite acceder de forma rápida y eficaz a la información almacenada en la base de datos. Esto incluye datos sobre mascotas, productos y proveedores lo que facilita tareas como gestor de pagos, control de inventarios y programación de compras, mejorando así la eficiencia en la administración contable de la veterinaria

Uso de base de datos propia para la gestión de información: Las bases de datos utilizadas de manera local con el servidor Wamp son muy útiles ya que funcionan de manera instantánea de manera local lo que hace el asistente es llamar al modelo para que realice una consulta como tal en la base de datos MySQL retornando en el propio Chatbot la información de esta

Mejora la experiencia del cliente o del vendedor para tomar buenas decisiones: La recopilación de información general para respuestas automáticas sobre problemas comunes ya sea para salud de la mascota o información detallada sobre una veterinaria es importante para ayudar a brindar respuestas tanto como para el vendedor como para el cliente basado con el modelo llama3 recopila información sobre enfermedades, medicamentos, etc. brindando respuesta como un veterinario profesional, también contando con la validación de que si se pregunta cosas fuera de contexto simule que no es adecuado en cuanto a un asistente de veterinaria

Cumplimiento Normativo y Seguridad de la Información: El chatbot actúa de forma diferente según el entorno en el que se encuentre. Si está en un entorno local, realizará consultas directamente a la base de datos MySQL, accediendo a información interna y exclusiva de la veterinaria. En cambio, si se encuentra en un entorno externo, donde no tiene acceso a la base de datos, responderá únicamente con la información disponible públicamente o recopilada de fuentes externas, siempre mencionando su origen. Esto garantiza el cumplimiento de normas de seguridad y protege la información confidencial de la veterinaria.

¿Cuáles son las estrategias metodológicas para la transformación digital en las prácticas contables aplicado en el campo de la veterinaria?

GESTION DE INFORMACION GENERAL

Ayuda a gestionar la información ya sea del cliente como del vendedor consultando información en la web sobre el cuidado de la mascota o cualquier pregunta referente a la veterinaria

BASE DE DATOS MYSQL

La base de datos es una respuesta clave la cual brinda información de manera inmediata sobre la veterinaria para que le ayude al asistente a recopilar su propia información

Corporación Universitaria Remington.
Ingeniería
Ingeniería de Sistemas.

Jhonatan Realpe y Eduin Cuasquer
Ing. Danny López Segura
Opción de Trabajo de grado Seminario
2025.

Agradecimientos

Agradecemos infinitamente a nuestro tutor Ing. Danny López Segura por el conocimiento brindado y el apoyo durante el transcurso del seminario, y a mis compañeros por el buen trabajo en equipo y sus experiencias y conocimiento compartido.

Tabla de Contenidos

Resumen.....	5
Palabra clave.....	5
Pregunta orientadora de la búsqueda	6
Metodología de búsqueda de la información.....	7
Sustentación teórica de la pregunta.....	8
Conclusiones.....	11
Referencias.....	292
Anexos	293

Resumen

El presente trabajo fue realizado con el fin de aprender sobre el uso de Chatbot con herramientas importantes como OLLAMA y sus modelos uno de ellos y el más rápido y eficaz es llama3 hemos realizado un código estructurado con Python para realizar la conexión a la base de datos propia la cual se integra con un servidor llamado Wamp esto funcionando de manera local para poder obtener los datos de manera instantánea, también se utilizó una página de HTML la cual nos sirve para mostrar el sistema de manera gráfica, utilizamos un script de tipo java Script para optimizar la respuesta de llama3

Este proyecto nos ha servido para profundizar el conocimiento sobre el impacto de la inteligencia artificial de hoy en día, ya que es una herramienta muy importante, debemos entender que hay que darle un buen uso para sacarle provecho a una herramienta muy útil como lo es la inteligencia artificial

Un Chatbot como el que realizamos ayuda a resolver problemas rápidos y de manera eficiente como un veterinario profesional esto ayudando a los clientes, en cuanto a los veterinarios con la ayuda de la base de datos brinda respuestas rápidas para poder obtener información sin necesidad de tener conocimiento de bases de datos ya que este programa ayuda a realizar las consultas con tan solo una palabra

El asistente lo hemos renombrado como LA LLAMA QUE LLAMA y nuestra veterinaria LOS PELUDITOS.

Problema: Una problemática muy buena para la que nuestro Chatbot puede ayudar es la necesidad de hacer consultas en la base de datos y no tener conocimientos del tema como también puede ser un problema que el veterinario no sea especializado y necesite saber información sobre alguna enfermedad

Objetivo: reflexionar sobre los métodos sistematizados en cuanto a la veterinaria en este caso con el uso de Chatbots y bases de datos

Metodología: Uso e implementación de chat para generar respuestas rápidas con base de datos propia

Reflexión:

Se crea un sistema de Chatbot que genera respuestas sobre dudas como un veterinario profesional, suma lista y hace cualquier función en la base de datos también contiene filtros para realizar consultas de la base de datos como por ejemplo de dicha mascota (LUCAS)

Llama3 realiza una búsqueda de la función y genera la lista de solamente esa mascota

Resultado es un análisis propio propuestas derivado de la fundamentación del estado del arte

Se realizo investigación de cómo funciona un chatbot, cómo funciona como tal el lenguaje de pythton y el lenguaje HTML, también se investigó sobre la utilización del api de OLLAMA con sus respectivos modelos para la utilización en el programa como también sen investigo sobre el uso de base de datos como es MYSQL con su servidor Wamp

Resultados esperados: Estratégias prácticas que incorporan el uso de las nuevas tecnologías y permiten mejorar la IF en el sector veterinario en cuanto a Chatbots

Impacto: Aprender a utilizar herramientas de inteligencia artificial como brindar soluciones con el impacto de la tecnología hoy en día y el uso de Chatbots como también las bases de datos para almacenar información de manera más rápida y segura

Palabras clave

(Chatbot, Base de datos Propia, Transformación digital, Automatización, Seguridad de la información)

Pregunta orientadora de la búsqueda

La Pregunta de la problemática ¿Cómo puede ayudar un sistema tecnológico a la utilización de base de datos y preguntas complicadas para el usuario?

Chatbot inteligente

El modelo de chat Bot realizado responde preguntas basándose a información de la web para una respuesta rápida y precisa tanto como para el veterinario como para el cliente que desee utilizarlo un ejemplo claro seria en caso de que un veterinario no tan especializado no sepa dar respuesta sobre alguna inquietud sobre la veterinaria, el asistente investiga a profundidad y ayuda al veterinario a dar respuestas mas claras

Base de datos

Captura información de la base de datos tanto de manera general como de manera especifica a esto solo tendrá acceso el veterinario no el cliente dando que se necesita conexión local para conectar el servidor Wamp con la Base de datos, en esta también se puede obtener la suma de dicho elemento seleccionado con tan solo describirle la opción por ejemplo “suma el total de mascotas que hay en la base de datos” esta consultara como una persona especializada en base de datos realiza una consulta y suma en la base de datos realizando funciones las cuales serán usadas por el sistema creado de Python

En caso de querer obtener datos de un solo elemento se deberá poner entre paréntesis dicho nombre ya sea de mascotas productos proveedores etc. por ejemplo “dame la suma total de la mascota llamada (DORLAN)” lo que consultará a la base de datos por solo dicho elemento, el sistema de Chatbot también muestra la lista de elementos seleccionados ya sea lista completa o lista de dichos elementos.

Metodología de búsqueda de la información

En este documento nos hemos guiado por la implementación de un Chatbot consultando análisis científicos investigando sobre el uso de herramientas como Apis Lenguajes servidores etc. Las palabras utilizadas o mas claras para este proyecto involucran tales como Chatbot, Base de datos Propia, Transformación digital, Automatización, Seguridad de la información.

Sustentación teórica de la pregunta

Nuestro sistema fue diseñado para brindar una herramienta que ayude a los veterinarios tanto como a clientes con la ayuda de respuestas sobre nuestra veterinaria la cual hemos nombrado como LOS PELUDITOS, nuestro Chatbot esta diseñado para responder cualquier tipo de pregunta solamente relacionada con veterinaria dado el caso que el usuario ingrese una pregunta no referente a veterinaria este devolverá una respuesta diciendo que no esta capacitado para responder cierto tipo de preguntas que no sean relacionadas con la veterinaria.

También contamos con una base de datos propia la cual obtiene información para mostrar los datos al Chatbot de manera fácil para el usuario de una manera profesional con tan solo una palabra como por ejemplo muéstrame, dime, indícame, etc. hemos agregado sinónimos hemos amaestrado a la IA de Llama3 para que nos devuelva datos exactos y que consulte dependiendo si quiere suma o lista completa

Utilizamos herramientas como los lenguajes Python y HTML, base de datos MySQL, servidor Wamp, Visual Studio Code, api de Ollama scripts de Java Script y entre otros.

Figuras y tablas

IMAGEN 1.

Estructura Del Código

En esta parte podemos visualizar como está dividido y organizado el código para su funcionamiento

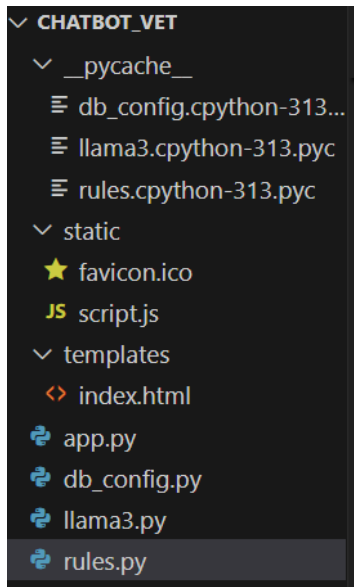


IMAGEN 3

Código de db_config.py

Identificamos nuestra conexión a la base de datos en este caso local usamos la librería `mysql.connector` que es la que usa Python para conectarse a la base de datos de MySQL

```
db_config.py > ...
1  import mysql.connector
2
3  def get_connection():
4      return mysql.connector.connect(
5          host="localhost",
6          user="root",
7          password="",
8          database="veterinaria"
9      )
10
```

IMAGEN 4

Código de rules.py

Rules.py hace la función de las reglas que se deben tener en cuenta para la funcionalidad de la base de datos como las consultas que el usuario va a generar con una sola palabra en ves de ejecutar las SQL como tal

```

rules.py > _
1 import re
2 from unicode import unidecode
3
4
5 def interpretar_pregunta(pregunta):
6     try:
7         pregunta = unidecode(pregunta.lower())
8         print(f"Pregunta procesada (sin tildes y en minúsculas): {pregunta}")
9
10        # ----- MASCOTAS -----
11        if re.search(r"\b(mascotas|animales|perros|gatos|mascotas llamadas|mascota)\b", pregunta):
12            nombre_match = re.search(r"\((.*?)\)", pregunta)
13            nombre = nombre_match.group(1).strip() if nombre_match else None
14            print(f"Nombre encontrado en la pregunta (entre paréntesis): {nombre}")
15
16            if re.search(r"\b(listar|mostrar|lista|revisar|revisa|ver|indicar|dame|muéstrame|listame|enseñar|consultar|visualizar|obtener|dime|enseñame|desplegar|tra",
17                pregunta):
18                if nombre:
19                    return f"SELECT * FROM mascotas WHERE nombre LIKE '%{nombre}%';"
20                return "SELECT * FROM mascotas;"
21
22            if re.search(r"\b(cuántas|cu[aa]ntas|cu[aa]ntos|n[ú]mero|numero|contar|cantidad|total|cantidad|suma|recuento|cuantificar|sumar|agrupar|contabilizar|enum",
23                pregunta):
24                if nombre:
25                    return f"SELECT COUNT(*) AS cantidad FROM mascotas WHERE nombre LIKE '%{nombre}%';"
26                else:
27                    return "SELECT COUNT(*) AS total_mascotas FROM mascotas;"
28
29        # ----- PROVEEDORES -----
30        if re.search(r"\b(proveedores|proveedor|vendedores|distribuidores)\b", pregunta):
31            nombre_match = re.search(r"\((.*?)\)", pregunta)
32            nombre = nombre_match.group(1).strip() if nombre_match else None
33            print(f"Nombre de proveedor encontrado: {nombre}")
34
35            if re.search(r"\b(listar|mostrar|lista|revisar|revisa|ver|indicar|dame|muéstrame|listame|enseñar|consultar|visualizar|obtener|dime|enseñame|desplegar|tra",
36                pregunta):
37                if nombre:
38                    return f"SELECT * FROM proveedores WHERE nombre LIKE '%{nombre}%';"
39                return "SELECT * FROM proveedores;"
40
41            if re.search(r"\b(cuántas|cu[aa]ntas|cu[aa]ntos|n[ú]mero|numero|contar|cantidad|total|cantidad|suma|recuento|cuantificar|sumar|agrupar|contabilizar|enum",
42                pregunta):
43                if nombre:
44                    return f"SELECT COUNT(*) AS cantidad FROM proveedores WHERE nombre LIKE '%{nombre}%';"
45                else:
46                    return "SELECT COUNT(*) AS total_proveedores FROM proveedores;"
47
48        # ----- PRODUCTOS -----
49        if re.search(r"\b(productos|producto|articulos|mercancia|item)\b", pregunta):
50            nombre_match = re.search(r"\((.*?)\)", pregunta)
51            nombre = nombre_match.group(1).strip() if nombre_match else None
52            print(f"Nombre de producto encontrado: {nombre}")
53
54            if re.search(r"\b(listar|mostrar|lista|revisar|revisa|ver|indicar|dame|muéstrame|listame|enseñar|consultar|visualizar|obtener|dime|enseñame|desplegar|tra",
55                pregunta):
56                if nombre:
57                    return f"SELECT * FROM productos WHERE nombre LIKE '%{nombre}%';"
58                return "SELECT * FROM productos;"
59
60            if re.search(r"\b(cuántas|cu[aa]ntas|cu[aa]ntos|n[ú]mero|numero|contar|cantidad|total|cantidad|suma|recuento|cuantificar|sumar|agrupar|contabilizar|enum",
61                pregunta):
62                if nombre:
63                    return f"SELECT COUNT(*) AS cantidad FROM productos WHERE nombre LIKE '%{nombre}%';"
64                else:
65                    return "SELECT COUNT(*) AS total_productos FROM productos;"
66
67        except Exception as e:
68            print(f"Error en la interpretación de la pregunta: {str(e)}")
69            return "Lo siento, algo salió mal. No pude procesar tu pregunta. ¿Puedes intentar [de]nuevo?"

```

IMAGEN 5

Código de index.html

Este código es el encargado de mostrar la interfaz del Chatbot en el navegador

```
<!doctype html>
<html lang="es">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>VETERINARIA LOS PELUDITOS</title>
  <link rel="stylesheet" href="https://unpkg.com/mvp.css">
  <style>

    body {
      font-family: Arial, sans-serif;
      margin: 0;
      padding: 0;
      background-color: #f0f0f5;
      display: flex;
      justify-content: center;
      align-items: center;
      min-height: 100vh;
      flex-direction: column;
      background: linear-gradient(135deg, #00b09b, #96c93d);
    }

    main {
      background: white;
      border-radius: 12px;
      width: 100%;
      max-width: 600px;
      padding: 20px;
      box-shadow: 0 4px 8px rgba(0, 0, 0, 0.1);
      box-sizing: border-box;
      overflow: hidden;
      text-align: center;
      position: relative;
    }

    h1 {
      font-size: 2em;
      color: #333;
      margin-bottom: 20px;
      font-weight: bold;
      animation: fadeIn 1.5s ease;
    }
  </style>

```

IMAGEN 5

Código De Script.Js

Es el encargado de la comunicación y optimización para la lógica de llama 3 como

El tipo de que si es de la base de datos o si es una pregunta normal que script debe ejecutar

```

const chat = document.getElementById("chat");
const form = document.getElementById("form");
const input = document.getElementById("pregunta");
const writingIndicator = document.getElementById("writing");

// Escape simple de caracteres peligrosos por si LLaMA devuelve cosas raras
function escapeHTML(str) {
  if (!str || typeof str !== "string") return "";
  return str.replace(/[<>"/]/g, function (m) {
    return {
      '&': '&amp;',
      '<': '&lt;',
      '>': '&gt;',
      '"': '&quot;',
      "'": '&#039;'
    }[m];
  });
}

form.addEventListener("submit", async (e) => {
  e.preventDefault();
  const pregunta = input.value.trim();
  if (!pregunta) return;

  appendMessage("Tú", escapeHTML(pregunta));
  input.value = "";
  input.disabled = true;
  writingIndicator.style.display = "flex";

  try {
    const res = await fetch("/chat", {
      method: "POST",
      headers: { "Content-Type": "application/json" },
      body: JSON.stringify({ pregunta }),
    });

    const data = await res.json();

    if (data && data.respuesta) {
      if (Array.isArray(data.respuesta)) {
        const listaDatos = data.respuesta.map(item => {
          return `
          <div>
            <strong>Nombre:</strong> ${escapeHTML(item.nombre)}<br>
            <strong>Especie:</strong> ${escapeHTML(item.especie)}<br>
            <strong>Raza:</strong> ${escapeHTML(item.raza)}<br>
            <strong>Edad:</strong> ${isNaN(item.edad) ? escapeHTML(item.edad) : parseInt(item.edad)}<br>
            <strong>Propietario:</strong> ${escapeHTML(item.propietario)}<br>
            <strong>Celular del Propietario:</strong> ${escapeHTML(item.celular_propietario)} || "N/A"<br>
            <strong>Historial Médico:</strong> ${escapeHTML(item.historial_medico)} || "N/A"<br>
          <br>
          </div>
          `;
        });
        listaDatos.join("");
        appendMessage("Bot", listaDatos, true);
      } else {
        appendMessage("Bot", escapeHTML(data.respuesta));
      }
    } else {
      appendMessage("Bot", "⚠️ No se recibió una respuesta del servidor.");
    }
  } catch (error) {
    console.error("❌ Error en la solicitud:", error);
    appendMessage("Bot", "❌ Hubo un problema al procesar tu solicitud.");
  }
}

```

IMAGEN 6

Código de Llama3.py

Aquí llamamos al modelo de llama3 para poderlo usar como Inteligencia Artificial usando el api de OLLAMA

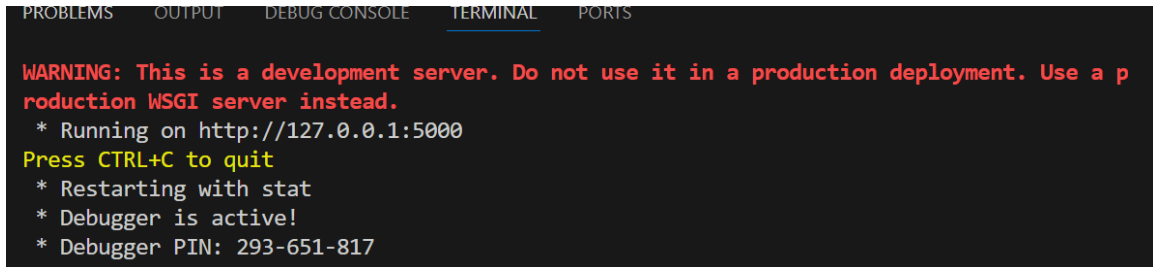
```
import requests

def preguntar_llama3(pregunta):
    try:
        response = requests.post(
            "http://localhost:11434/api/generate",
            json={
                "model": "llama3",
                "prompt": pregunta,
                "stream": False
            }
        )
        response.raise_for_status()
        data = response.json()
        return data['response'].strip()
    except requests.exceptions.RequestException as e:
        print(f"Error al conectarse con Ollama: {e}")
        return "Hubo un error al conectar con la IA."
```

IMAGEN 7

Correr El Programa

AL presionar play en nuestro código de app.py empezara a correr el sistema mostrando una consola de logs para verificar que todo esté funcionando correctamente



```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
WARNING: This is a development server. Do not use it in a production deployment. Use a p
roduction WSGI server instead.
* Running on http://127.0.0.1:5000
Press CTRL+C to quit
* Restarting with stat
* Debugger is active!
* Debugger PIN: 293-651-817
```

IMAGEN 8

Interfaz De Chat Bot

Al entrar al localhost o ruta local podemos visualizar la interfaz de nuestro chat bot una interfaz intuitiva y fácil de usar como asistente

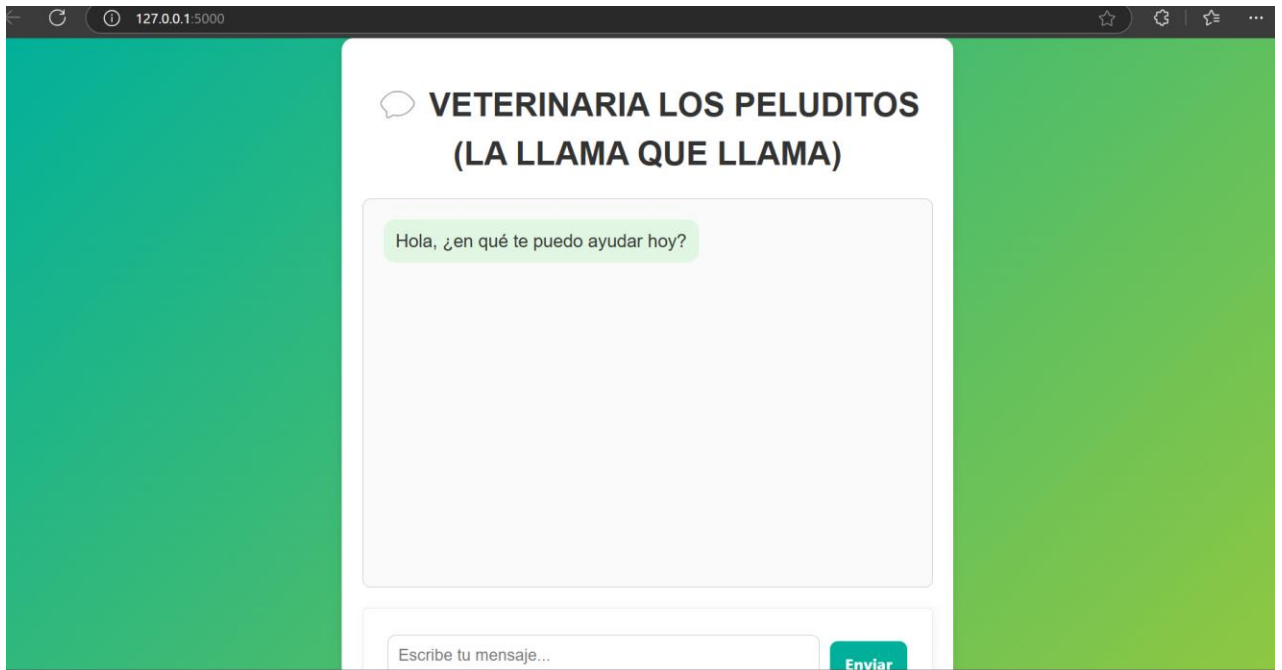


IMAGEN 8

Saludos

Nuestra inteligencia artificial detecta que es un saludo y lo responde con una bienvenida

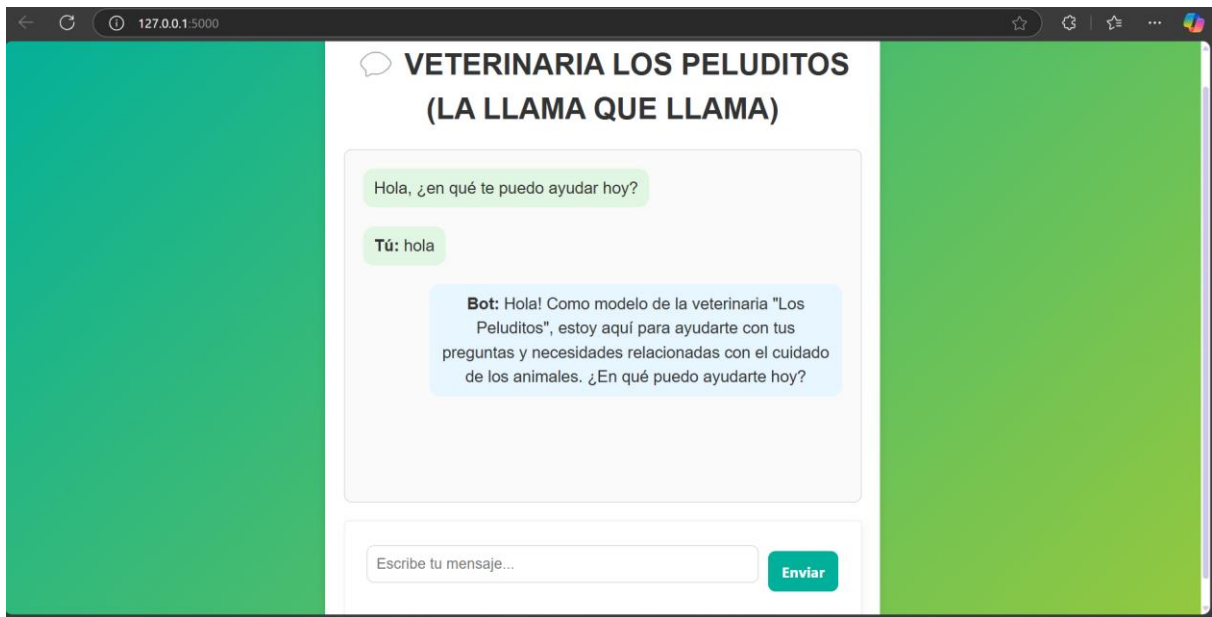


IMAGEN 9

Investigación

Consulta información relevante basado en veterinaria llama3 detecta que no es una consulta y la investiga

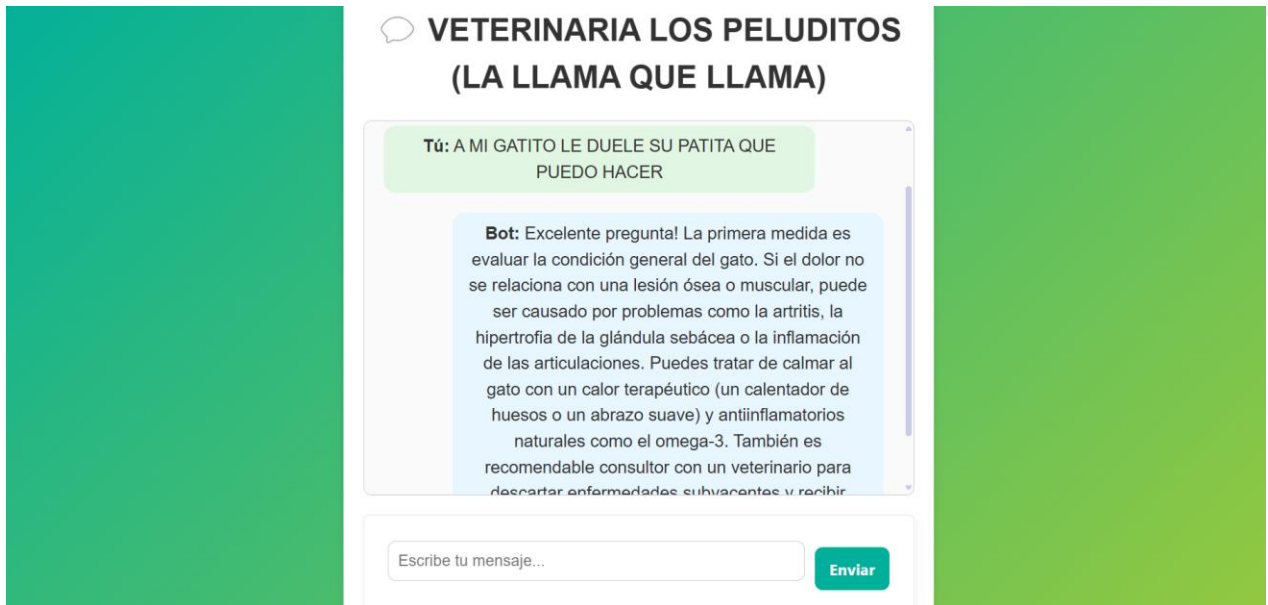


IMAGEN 10

Negaciones

En caso de que el usuario pregunte sobre información que no sea relacionada esta manda una negación

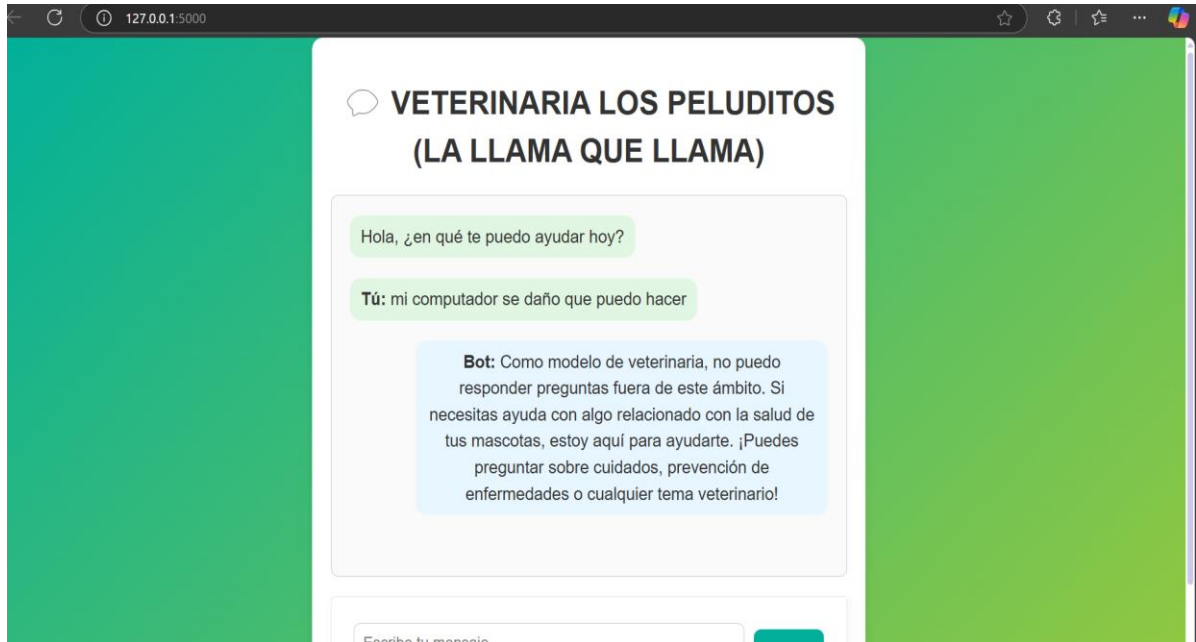


IMAGEN 11

Consultas de base de datos

la parte mas importante es la de nuestra base de datos con la cual podemos hacer varios tipos de consulta ya sea por elemento o en general tanto listar como también sumar ya sea mascotas, productos proveedores, etc.

IMAGEN 12

listar

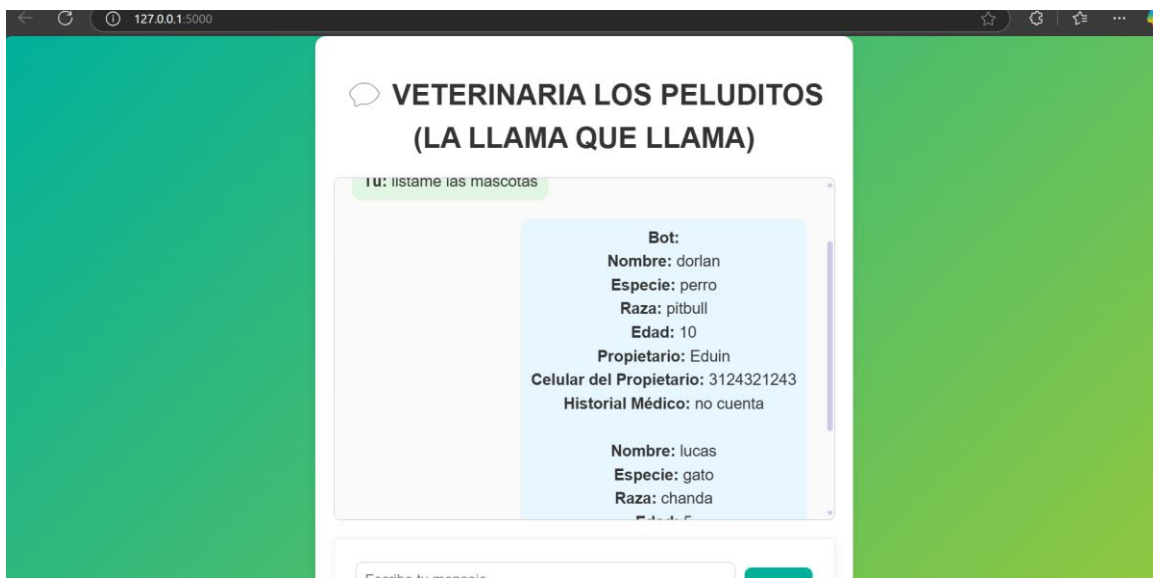


IMAGEN 13

Listar por elemento específico con paréntesis

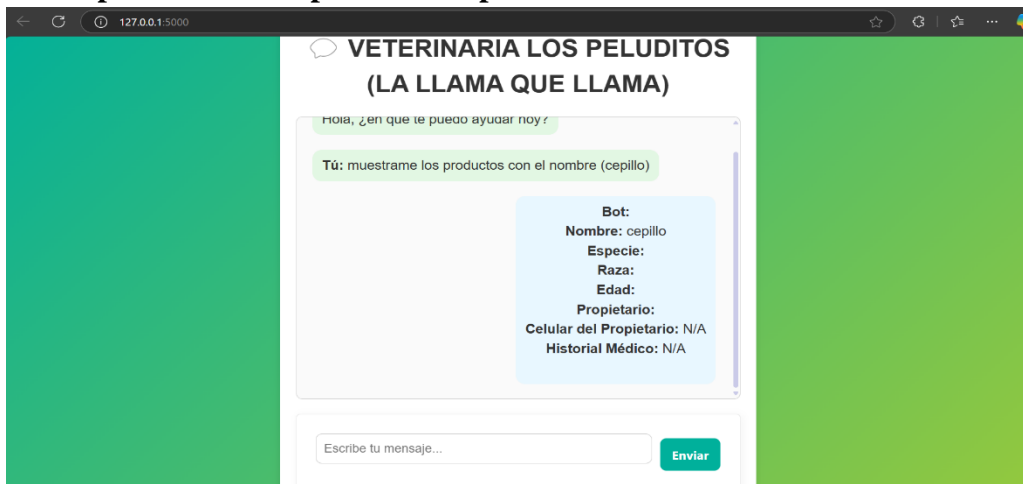


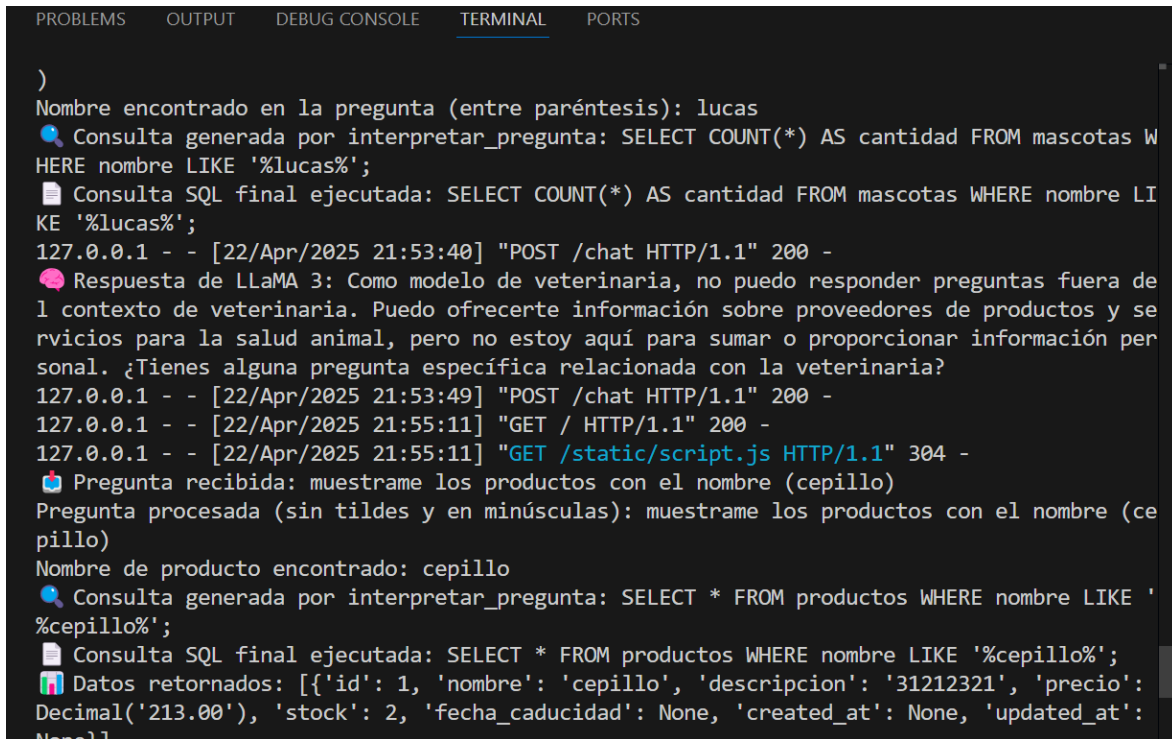
IMAGEN 14**Sumar**

IMAGEN 15**Base de datos correspondiente a la veterinaria**

IMAGEN 16

Logs

En el terminal de donde se esta ejecutando app.py muestra los logs de lo que interpreta el chatbot



```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
)
Nombre encontrado en la pregunta (entre paréntesis): lucas
🔍 Consulta generada por interpretar_pregunta: SELECT COUNT(*) AS cantidad FROM mascotas W
HERE nombre LIKE '%lucas%';
📄 Consulta SQL final ejecutada: SELECT COUNT(*) AS cantidad FROM mascotas WHERE nombre LI
KE '%lucas%';
127.0.0.1 - - [22/Apr/2025 21:53:40] "POST /chat HTTP/1.1" 200 -
🗨 Respuesta de LLaMA 3: Como modelo de veterinaria, no puedo responder preguntas fuera de
l contexto de veterinaria. Puedo ofrecerte información sobre proveedores de productos y se
rvicios para la salud animal, pero no estoy aquí para sumar o proporcionar información per
sonal. ¿Tienes alguna pregunta específica relacionada con la veterinaria?
127.0.0.1 - - [22/Apr/2025 21:53:49] "POST /chat HTTP/1.1" 200 -
127.0.0.1 - - [22/Apr/2025 21:55:11] "GET / HTTP/1.1" 200 -
127.0.0.1 - - [22/Apr/2025 21:55:11] "GET /static/script.js HTTP/1.1" 304 -
📄 Pregunta recibida: muestrame los productos con el nombre (cepillo)
Pregunta procesada (sin tildes y en minúsculas): muestrame los productos con el nombre (ce
pillo)
Nombre de producto encontrado: cepillo
🔍 Consulta generada por interpretar_pregunta: SELECT * FROM productos WHERE nombre LIKE '
%cepillo%';
📄 Consulta SQL final ejecutada: SELECT * FROM productos WHERE nombre LIKE '%cepillo%';
📄 Datos retornados: [{'id': 1, 'nombre': 'cepillo', 'descripcion': '31212321', 'precio':
Decimal('213.00'), 'stock': 2, 'fecha_caducidad': None, 'created_at': None, 'updated_at':
None}]
```

Conclusiones.

En este trabajo aprendimos sobre la importancia de la inteligencia artificial y su eficacia en la vida cotidiana, como también a poder usar Apis y los Chatbots de varias empresas para la investigación de nuestro trabajo, hemos usado varios modelos sobre Chatbots empresariales que nos han servido de ayuda.

Esto es una estrategia que es innovadora ya que facilita respuestas tanto como para el vendedor como para el cliente, siendo una ayuda muy útil en el campo de la tecnología ya que da respuestas rápidas, contantes y personalizadas.

Referencias

Asistente DeepSeek y Chatgpt

<https://sider.ai/apps/lp->

[chatgpt?source=b&p1=chatgpt&p2=search&msclkid=5ab997810a671c7bc58dff8950943](https://sider.ai/apps/lp-chatgpt?source=b&p1=chatgpt&p2=search&msclkid=5ab997810a671c7bc58dff8950943)

28a

Investigación sobre Apis

<https://lureodigital.com/todo-lo-que-necesitas-saber-sobre-las-apis-definicion-usos-y->

[tipos/](https://lureodigital.com/todo-lo-que-necesitas-saber-sobre-las-apis-definicion-usos-y-tipos/)

Guía para entender a los Chatbots

<https://centromexico.digital/wp->

[content/uploads/2023/03/07022023KTBO_CMD_Whitepapper_Guia-para-entender-a-](https://centromexico.digital/wp-content/uploads/2023/03/07022023KTBO_CMD_Whitepapper_Guia-para-entender-a-)

[los-chatbots.pdf](https://centromexico.digital/wp-content/uploads/2023/03/07022023KTBO_CMD_Whitepapper_Guia-para-entender-a-los-chatbots.pdf)