



**universidad  
de león**



# **Escuela de Ingenierías Industrial, Informática y Aeroespacial**

## **GRADO EN INGENIERÍA INFORMÁTICA**

**Desarrollo de un Chatbot para Recomendaciones  
Dietéticas Personalizadas**

**Autor: Alicia Gómez Pascual**



# Índice

<b>Introducción.....</b>	<b>3</b>
<b>Objeto de estudio y problema a resolver.....</b>	<b>4</b>
<b>Tecnologías y herramientas utilizadas.....</b>	<b>5</b>
<b>Cómo se resuelve el problema y funcionamiento de la aplicación.....</b>	<b>7</b>
Configuración inicial:.....	7
Recopilación de datos del usuario:.....	7
Interacción con el Chatbot:.....	7
Recomendación de alimentos:.....	8
<b>Estudio DAFO de la aplicación.....</b>	<b>9</b>
Debilidades:.....	9
Amenazas:.....	9
Fortalezas:.....	10
Oportunidades:.....	10
<b>Conclusión.....</b>	<b>12</b>
<b>Bibliografía.....</b>	<b>13</b>



# Introducción

Este proyecto está enfocado en el desarrollo de un chatbot para ofrecer recomendaciones dietéticas personalizadas, dirigido a aquellas personas que necesitan gestionar sus hábitos alimenticios según sus características personales, restricciones y objetivos de salud. En un contexto donde las enfermedades relacionadas con una alimentación inadecuada están en aumento, la demanda de sistemas que proporcionen orientación nutricional precisa y ajustada a las necesidades individuales es cada vez mayor. El chatbot desarrollado en este proyecto tiene la capacidad de adaptarse a factores como las alergias alimentarias, los objetivos dietéticos y el estilo de vida del usuario, proporcionando recomendaciones personalizadas y dinámicas que mejoran la experiencia y efectividad en el seguimiento de dietas saludables. A través de una interfaz web interactiva y el uso de tecnologías de procesamiento de lenguaje natural, se busca resolver el desafío de ofrecer soluciones dietéticas que se ajusten de forma precisa y continua a las necesidades del usuario.



## Objeto de estudio y problema a resolver

El objeto central de este proyecto es el desarrollo de un chatbot especializado en dietas, cuyo objetivo es ofrecer recomendaciones dietéticas personalizadas basadas en las características y necesidades individuales de cada usuario. El sistema interactúa con los usuarios a través de una interfaz web donde se ingresan datos como peso, altura, alergias, nivel de actividad física y objetivos personales (adelgazar, aumentar masa muscular o mantenerse saludable). Con esta información, el chatbot genera planes de alimentación adaptados y seguros, teniendo en cuenta las restricciones alimentarias específicas.

Uno de los principales problemas que aborda este proyecto es la dificultad para acceder a recomendaciones dietéticas personalizadas, especialmente cuando existen restricciones alimenticias como alergias alimentarias. Las personas con necesidades dietéticas especiales, como intolerancias o alergias, a menudo tienen dificultades para encontrar planes de alimentación adecuados y seguros. Además, mantener la constancia en el seguimiento de una dieta puede ser complicado, sobre todo cuando no se cuenta con una fuente de orientación continua y ajustada a la evolución del usuario. Este proyecto busca solucionar este problema al proporcionar un sistema interactivo que adapta las recomendaciones a medida que cambian las condiciones del usuario, asegurando una experiencia personalizada y efectiva.

Este problema es particularmente relevante dado el creciente número de personas que padecen alergias alimentarias y que requieren dietas específicas para evitar reacciones adversas. La falta de información precisa y personalizada puede llevar a situaciones riesgosas para la salud, como reacciones alérgicas graves. Además, la falta de una guía constante y adaptada al usuario hace que muchas personas abandonen sus esfuerzos por mantener una alimentación saludable. No abordar esta situación puede aumentar la incidencia de problemas de salud relacionados con la nutrición, como desnutrición, obesidad o complicaciones derivadas de alergias alimentarias.



# Tecnologías y herramientas utilizadas

Para el desarrollo de este proyecto se emplearon diversas tecnologías y herramientas que permitieron crear un sistema eficiente, interactivo y adaptable a las necesidades del usuario. El lenguaje de programación principal utilizado fue Python, debido a su flexibilidad y amplio soporte para la integración de diferentes módulos y librerías. Aproximadamente el 97% del código está escrito en Python, lo que permitió manejar el procesamiento de datos, la lógica del chatbot y la comunicación con la API externa.

En cuanto a la automatización de tareas y la gestión del entorno, se empleó Shell scripting (un 2.2% del proyecto), que facilitó la ejecución de scripts y la automatización de ciertos procesos del sistema. Para garantizar la ejecución consistente de la aplicación en diferentes entornos, se utilizó Docker. A través de un Dockerfile, se crearon imágenes de Docker que aseguraron que el entorno de ejecución fuera el mismo, independientemente de la máquina o sistema operativo utilizado.

El código fuente del proyecto se gestionó mediante Git, un sistema de control de versiones ampliamente utilizado que permitió hacer un seguimiento adecuado de los cambios y colaborar de forma eficiente. El repositorio del proyecto se encuentra alojado en GitHub [1], lo que permitió una gestión centralizada del código y el acceso remoto por parte de los desarrolladores. Además, para la creación y ejecución de Docker, se automatizó el proceso mediante un Makefile, lo que facilitó la construcción de contenedores y la puesta en marcha de la aplicación con pocos comandos.

El procesamiento de lenguaje natural (PLN) en el sistema se lleva a cabo mediante la API de Cohere, que permite generar respuestas de forma dinámica y coherente según las consultas del usuario. Esta API fue elegida por su accesibilidad gratuita y su rendimiento similar al de otras alternativas como la de OpenAI. Asimismo, para mejorar las recomendaciones de recetas en función de las preferencias y restricciones alimenticias del usuario, se implementó RAG (Retrieval-Augmented Generation), que permite hacer recomendaciones más precisas al buscar y filtrar alimentos según las alergias y otros parámetros proporcionados.

El proyecto también hizo uso de diversas librerías de Python, como pandas para la manipulación de datos, tqdm para mostrar barras de progreso durante procesos largos, y numpy para el manejo de datos numéricos. Otras librerías clave fueron faiss, que facilita la búsqueda rápida en grandes volúmenes de datos, y sentence\_transformers, utilizada para convertir texto en vectores y calcular similitudes semánticas entre ellos. Además, se utilizaron herramientas como kagglehub para la integración de datasets y ast para la manipulación de las respuestas dadas por los agentes.

La interfaz web interactiva se desarrolló con Streamlit, que permitió crear una experiencia de usuario fluida, permitiendo la entrada de datos personales y la



interacción con el chatbot. Para garantizar que el código estuviera libre de errores, se implementaron pruebas unitarias utilizando pytest, y se evaluó la cobertura de las pruebas mediante coverage, asegurando que todas las partes del sistema estuvieran correctamente verificadas.

En conjunto, estas tecnologías y herramientas proporcionaron una infraestructura sólida y flexible para la creación de un chatbot efectivo, capaz de ofrecer recomendaciones dietéticas personalizadas y adaptadas a las necesidades de cada usuario.



# Cómo se resuelve el problema y funcionamiento de la aplicación

La aplicación resuelve el problema de la falta de recomendaciones dietéticas personalizadas, especialmente para personas con restricciones alimenticias como alergias, mediante un sistema interactivo que adapta las recomendaciones en función de los datos introducidos por el usuario. El funcionamiento de la aplicación se puede dividir en las siguientes fases:

## Configuración inicial:

Para facilitar la implementación en diferentes entornos, se utiliza un makefile que crea y ejecuta un contenedor Docker. Esto asegura que la aplicación se ejecute de manera consistente, independientemente de la plataforma utilizada.

Durante la configuración inicial (setup), se crea una base de datos con miles de recetas. Esta base de datos contiene información crucial sobre los alérgenos presentes en los alimentos, lo que es fundamental para el sistema de recomendaciones y la personalización de las dietas.

## Recopilación de datos del usuario:

Al ejecutar la aplicación, se genera una página web donde el usuario debe introducir información personal como sus alergias, peso, altura, edad, y nivel de actividad física. Además, se le pregunta sobre su objetivo en relación a su dieta: si desea adelgazar, aumentar masa muscular, o simplemente mantenerse sano.

Un agente denominado "verify\_data\_agent" es el encargado de verificar la coherencia de los datos introducidos por el usuario. Si los datos no son lógicos (por ejemplo, un IMC demasiado bajo), el sistema genera un error y solicita correcciones.

Una vez validados los datos, el "def\_user\_agent" crea una descripción textual detallada del usuario y sus necesidades alimenticias, que se mostrará en la interfaz de la aplicación, proporcionando una visión general de sus objetivos y restricciones.

## Interacción con el Chatbot:

Después de completar el formulario, el usuario ve una página que presenta su descripción a la izquierda y un chat a la derecha, donde puede hacer preguntas o modificar su perfil.



Cuando el usuario introduce una pregunta en el chat, el sistema evalúa si la pregunta aporta nueva información sobre su perfil. Para ello, se utiliza el agente "detect\_change\_data\_agent".

Si la pregunta incluye nueva información (por ejemplo, si el usuario menciona un cambio en su peso, como "mi peso es en realidad 58"), dicha información es verificada nuevamente a través de "verify\_data\_agent". Si los datos son coherentes, el sistema actualiza el perfil del usuario, y la descripción se reescribe usando "def\_user\_agent".

Si la pregunta no cambia el perfil del usuario, el sistema pasa la consulta al agente "recommendation\_agent" para generar recomendaciones.

## **Recomendación de alimentos:**

El "recommendation\_agent" emplea una técnica de RAG (Retrieval-Augmented Generation) para buscar en la base de datos de alimentos creada durante la configuración inicial. Esta técnica permite filtrar los alimentos según las alergias del usuario, generando recomendaciones alimenticias personalizadas que se ajustan a las necesidades del usuario.

El agente busca también alimentos que mejor se adapten a las indicaciones del usuario, ya sea en términos de objetivos dietéticos o restricciones alimenticias.





# Estudio DAFO de la aplicación

## Debilidades:

**Dependencia de la API de Cohere:** La aplicación depende de la API de Cohere para el procesamiento del lenguaje natural, lo que puede ser una debilidad en caso de que esta API cambie sus condiciones o deje de ser gratuita. Además, el rendimiento de Cohere es inferior al de otras APIs similares, lo que puede afectar la velocidad y la calidad de las respuestas.

**Dependencia de datos precisos:** La calidad de las recomendaciones está directamente vinculada a la precisión de los datos ingresados por los usuarios. Si los datos no son coherentes o están mal introducidos, la aplicación puede generar resultados incorrectos o imprecisos.

**Interacción inicial limitada:** La aplicación se basa en un formulario inicial para recopilar los datos del usuario, lo que puede ser visto como una limitación por aquellos usuarios que prefieren una interacción más flexible o dinámica a lo largo del uso.

## Amenazas:

**Competencia:** Existen muchas aplicaciones similares en el mercado que ofrecen planes de alimentación personalizados, algunas con bases de datos más grandes, algoritmos más avanzados y funcionalidades adicionales. La competencia podría dificultar la adopción de la aplicación, especialmente si las soluciones existentes son más completas o mejor conocidas.

**Cambios en la API de Cohere:** La API de Cohere es un recurso clave para el procesamiento de lenguaje natural en la aplicación. Si en el futuro se modifican las condiciones de uso o la API deja de ser gratuita, el proyecto podría enfrentar problemas técnicos y financieros, lo que obligaría a buscar alternativas más costosas o complicadas.

**Desactualización de la base de datos:** Si la base de datos de alimentos no se actualiza o amplía, las recomendaciones podrían volverse menos relevantes, limitando la efectividad de la aplicación. Esto podría llevar a la insatisfacción de los usuarios y a la disminución de la fidelización.

**Resistencia de los usuarios a la tecnología:** Algunos usuarios pueden sentirse incómodos con la interacción a través de un chatbot o con el formulario inicial. Esto podría ser una barrera para la adopción de la aplicación, especialmente entre aquellos que prefieren interfaces más tradicionales o personalizadas.



## Fortalezas:

**Personalización avanzada:** La aplicación permite crear un perfil detallado del usuario, incluyendo datos como alergias, objetivos nutricionales, nivel de actividad física, entre otros. Esta personalización ayuda a ofrecer recomendaciones de dieta altamente ajustadas a las necesidades individuales.

**Interacción mediante chatbot:** La integración de un sistema de chat interactivo permite a los usuarios hacer preguntas y recibir recomendaciones de alimentos de manera dinámica y personalizada. Esta funcionalidad facilita la comunicación constante con el sistema.

**Verificación de datos:** La aplicación incluye un agente que verifica la coherencia de los datos ingresados por el usuario, lo que reduce la probabilidad de errores en las recomendaciones y asegura que los planes de alimentación sean apropiados y adecuados para el usuario.

**Uso de la API de Cohere:** El uso de la API de Cohere para el procesamiento del lenguaje natural es una ventaja, ya que es una solución gratuita que facilita el desarrollo de los agentes encargados de la verificación de datos, la definición del perfil y la generación de recomendaciones.

## Oportunidades:

**Expansión de la base de datos de alimentos:** Existe una oportunidad significativa de ampliar la base de datos de recetas y alimentos, lo que permitiría ofrecer recomendaciones más completas y variadas. Incluir más alimentos, recetas internacionales y opciones dietéticas especializadas podría atraer a un público más amplio.

**Ampliación del soporte multilingüe:** La aplicación podría ser adaptada a diferentes idiomas, lo que permitiría ampliar su alcance a un público global. La expansión a varios idiomas aumentaría su accesibilidad y potencial de usuarios en diferentes países.

**Integración con dispositivos wearables:** Incorporar dispositivos como relojes inteligentes o pulseras de actividad podría proporcionar datos más precisos sobre la actividad física y la salud del usuario, lo que permitiría ofrecer recomendaciones aún más personalizadas.

**Diversificación en opciones dietéticas:** La aplicación podría incluir opciones dietéticas más variadas, como menús veganos, vegetarianos o adaptados a restricciones religiosas (por ejemplo, dietas halal o kosher), lo que ampliaría el espectro de usuarios.



**Planes de alimentación a largo plazo:** Ofrecer la opción de planes alimenticios mensuales o de largo plazo podría ser una oportunidad para mejorar la experiencia del usuario, proporcionando una estructura más sólida para aquellos que deseen un seguimiento continuado de su dieta.



## Conclusión

En conclusión, el desarrollo de este chatbot de recomendaciones dietéticas personalizadas ha permitido abordar una problemática creciente en la sociedad actual: la dificultad de acceder a planes de alimentación adecuados y seguros, especialmente para personas con restricciones alimentarias o necesidades específicas. Gracias al uso de tecnologías avanzadas como Python, la API de Cohere para el procesamiento del lenguaje natural, y la integración de un sistema interactivo basado en un chatbot, se ha logrado ofrecer una solución eficaz que mejora la interacción y personalización del usuario en el ámbito nutricional. Si bien la aplicación presenta algunas limitaciones y amenazas, como la dependencia de la API utilizada y la necesidad de mantener actualizada la base de datos de recetas, las oportunidades de expansión y mejora continúan siendo amplias, lo que abre nuevas posibilidades para su crecimiento y adaptación a las necesidades cambiantes del público.



## Bibliografía

- [1] Alicia Gómez Pascual. (2025). SIBI\_chatbot\_diets [Repositorio de GitHub]. GitHub. [https://github.com/Alicia0629/SIBI\\_chabot\\_diets](https://github.com/Alicia0629/SIBI_chabot_diets)
- [2] Cohere. (2025). Cohere API Documentation. <https://cohere.ai/docs/>
- [3] Docker Inc. (2025). Docker Documentation. <https://docs.docker.com/>
- [4] Kaggle. (2025). Kaggle Datasets. <https://www.kaggle.com/datasets>
- [5] Streamlit. (2025). Streamlit Documentation. <https://docs.streamlit.io/>