

Universidad Internacional de La Rioja

Facultad de Ciencias de la Salud

Máster Universitario en Bioinformática

Desarrollo de un algoritmo predictivo para la personalización de estilos de vida basado en genómica y biométrica

|  |  |
| --- | --- |
| Trabajo fin de Estudio presentado por: | Maximiliano Hernández Sahuquillo y David Fernández Martín |
| Tipo de trabajo: | 2 |
| Director/a: | Dr. José Arturo Mora Soto |
| Fecha: | 14 de marzo de 2024 |

Resumen

El presente TFM tiene como objetivo desarrollar un algoritmo predictivo para la personalización de estilos de vida basado en datos genómicos y biométricos. El algoritmo estará diseñado para generar recomendaciones personalizadas de estilo de vida para cada individuo, teniendo en cuenta su predisposición genética a ciertas enfermedades, su respuesta al ejercicio y su respuesta a diferentes tipos de dieta.

**Palabras clave:** estilo de vida, genómica, biométrica

Abstract

This TFM aims to develop a predictive algorithm for lifestyle personalisation based on genomic and biometric data. The algorithm will be designed to generate personalised lifestyle recommendations for each individual, taking into account their genetic predisposition to certain diseases, their response to exercise and their response to different types of diet.

**Keywords**: lifestyle, genomics, biometrics

Índice de contenidos

[1. Introducción 8](#_Toc165992245)

[1.1. Justificación 8](#_Toc165992246)

[1.2. Planteamiento del problema 10](#_Toc165992247)

[1.3. Objetivos 10](#_Toc165992248)

[2. Marco teórico 11](#_Toc165992249)

[3. Metodología 13](#_Toc165992250)

[3.1. Recolección de datos y preparación del modelo 13](#_Toc165992251)

[Preparación de los datos 13](#_Toc165992252)

[Entrenamiento del modelo 13](#_Toc165992253)

[3.2. Desarrollo del prototipo de la web 13](#_Toc165992254)

[Consideraciones para el desarrollo de la web 13](#_Toc165992255)

[Pasos para el desarrollo de nuestra web 14](#_Toc165992256)

[Pruebas de validación 17](#_Toc165992257)

[3.3. Esquema y funcionamiento formal del RAG 17](#_Toc165992258)

[Funcionamiento de llama3.py. 17](#_Toc165992259)

[3.3.2 Integración de Tecnologías: 18](#_Toc165992260)

[3.3.3 Detalles del Código: 19](#_Toc165992261)

[4. Resultados 21](#_Toc165992262)

[4.1. “Título 2” del menú de estilos 21](#_Toc165992263)

[“Título 3” del menú de estilos 21](#_Toc165992264)

[“Título 3” del menú de estilos 21](#_Toc165992265)

[5. Discusión 21](#_Toc165992266)

[5.1. “Título 2” del menú de estilos 21](#_Toc165992267)

[“Título 3” del menú de estilos 21](#_Toc165992268)

[“Título 3” del menú de estilos 21](#_Toc165992269)

[6. Conclusiones 21](#_Toc165992270)

[6.1. “Título 2” del menú de estilos 21](#_Toc165992271)

[“Título 3” del menú de estilos 21](#_Toc165992272)

[“Título 3” del menú de estilos 21](#_Toc165992273)

[Anexo A. Diagramas Mermaid del trabajo. 22](#_Toc165992274)

[Referencias bibliográficas 23](#_Toc165992275)

Índice de figuras

[Figura 1. “Figuras” del menú de estilos. (Elaboración propia) 9](#_Toc20304756)

Índice de tablas

[Tabla 1. “Tablas” del menú de estilos 8](#_Toc20304757)

# Introducción

El presente Trabajo Fin de Máster (TFM) pretende desarrollar y poner en práctica un servicio web basado en datos biométricos, genéticos y distintos aspectos relacionados con la salud los cuales van a ser procesados por varios LLM (Large Language Model) para su correcta implementación. El motivo que condujo a esta idea es el de aportar un sistema que personalice los estilos de vida de cada paciente.

En las próximas páginas se introducirá este trabajo, primero se abordarán los motivos que llevaron a escoger esta temática, seguido de la introducción del problema a resolver y, por último, se plantearán el objetivo principal y objetivos secundarios a alcanzar.

## Justificación

En los últimos años, el campo de la inteligencia artificial ha sufrido un avance vertiginoso, particularmente con respecto al procesamiento del lenguaje natural (NLP) y el desarrollo de modelos de lenguaje de gran tamaño (LLM)(1). Estos consiguieron gran notoriedad y presencia en el sector médico, gracias a sus múltiples aplicaciones, desde el procesamiento de bastas cantidades de datos hasta poder interpretar anotaciones y datos clínicos para generar reportes adecuados.

En estudios recientes acerca del rendimiento de bots de chat basados en LLMs al responder preguntas médicas, han demostrado una impresionante cantidad de aciertos con respecto al conocimiento sobre bases de datos médicas. Esto es debido ya que en primer lugar, poseen suficientes conocimientos médicos para aplicaciones en medicina, lo que podría provocar un cambio de paradigma en la prestación de asistencia sanitario (2). Pero persiste una considerable inquietud sobre la preocupación de que los LLM puedan ser manipulados maliciosamente para generar contenidos dañinos o engañosos (3).

Permitiendo que el proceso de cribado de la literatura (que se vuelve excesivamente laboriosa debido a que pueden existir miles de artículos con respecto a una patología o un gen) se vuelva más rápido y eficaz, permitiendo resumir la literatura científica y dejando a los investigadores y médicos la información de manera accesible con respecto al esfuerzo realizado (4).

A su vez en otros campos relacionados con la salud como la nutrición, numerosos estudios han contribuido al desarrollo de diversos métodos de recomendación de alimentos (5,6). Estos enfoques mediante el uso de LLM no sólo simplifican el proceso de elección de alimentos, sino que también promueven la adopción de hábitos sostenibles y saludables entre los usuarios. El poder de estas recomendaciones reside en su capacidad para integrar la ciencia de la nutrición en las necesidades de la población o de subpoblaciones, ofreciendo una guía intuitiva y práctica para las personas que desean tomar decisiones dietéticas informadas.

No obstante, los servicios convencionales de recomendación de alimentos orientados a la nutrición han encontrado limitaciones a la hora de comprender exhaustivamente la intrincada interacción entre la salud y el bienestar de las personas, que abarca parámetros fisiológicos, actividad física y calidad del sueño, y de alinearlos con necesidades nutricionales personalizadas.

Concretamente, estos sistemas suelen tener dificultades para adaptar sus sugerencias alimentarias a las necesidades nutricionales específicas de cada persona. La variabilidad inherente a la forma en que la nutrición afecta a los individuos suscita preocupación por la ausencia de una auténtica personalización, además otra carencia importante es la falta de interactividad.

Es en este punto donde se llega al objeto de estudio de este Trabajo Fin de Máster, el de como implementar un servicio sencillo e intuitivo, pero a la vez complejo, que permite a los usuarios que accedan a la web obtener recomendaciones personalizadas desde dudas sobre ciertas patologías que puedan llegar a presentar (obviando el hecho de que la asistencia a un profesional de la salud será necesaria) hasta alimentos o cambios que podrán realizar en función de distintos objetivos que posean.

## Planteamiento del problema

El objeto de estudio de este TFM, como se ha presentado en el apartado anterior es desarrollar un sistema que personalice los estilos de vida en función de la genética y la biometría, ya que recomendaciones generales sobre dieta y ejercicio son útiles, pero no siempre son efectivas para todas las personas.

## Objetivos

Como ya se ha adelantado, la propuesta consiste en desarrollar una web basada en varios LLM que permita el ingreso de los interesados de sus datos, biométricos, genéticos y otras dudas médicas que presenten (se responderá orientativamente).

**Objetivo principal:** desarrollar un sistema de personalización de estilos de vida basado en genómica y biometría. Este sistema tiene el potencial de mejorar la salud y el bienestar de las personas al proporcionarles recomendaciones personalizadas sobre dieta, ejercicio y otros aspectos del estilo de vida.

Objetivos secundarios:

* Mejorar la salud y el bienestar de las personas gracias al sistema de personalización de estilos de vida
* Reducir los costos de atención médica
* Empoderar a las personas para que tomen el control de su salud

# Marco teórico

No habría una fuente bibliográfica la cual haya significado una mayor influencia sobre cómo estructurar el framework o el proceso de realizar la web, sino que se ha ido obteniendo la información de distintas publicaciones u otras referencias.

Teniendo como referencia para proceder a herramientas como las publicadas ChatDiet (7) o Dietos (5), los cuales permiten comprender la manera de organizar el framework con respecto al uso de LLM para personalizar distintas necesidades del cliente (en el caso de ambas herramientas mencionadas en el campo de la nutrición).

Pero también se sigue un procedimiento similar en otros trabajos que proponen frameworks con LLM para datos genéticos (8).

Es decir, que para uno de los dos LLM que se procede a implementar en base a la lectura de documentos, la estructura básica es la misma seguida independientemente del ámbito relacionado con la salud escogido. Por lo que el primer paso será recopilar una gran cantidad de documentos de interés o extraerlos directamente de las webs necesarias, que permitirán al LLM generar respuestas en base a los diagnósticos o preguntas que proporcione el usuario.

Para el otro LLM, se utilizan los datos pertinentes de bases de datos como NHANES u otras similares, los cuales permitirán ser codificados para su posterior uso para obtener distintas estadísticas y porcentajes pertinentes para mostrar al usuario.

Así aportando una información más exacta acerca de por ejemplo “las posibilidades de padecer X patología o enfermedad” en base a los datos ingresados por el usuario como información sobre distintas dudas que posea acerca del diagnóstico proporcionado o acerca de otros ámbitos relacionados con la salud. Todo esto acompañado de las figuras necesarias para mostrar la información de la manera más accesible posible.

# Metodología

## Recolección de datos y preparación del modelo

### Preparación de los datos

En primera instancia, se recopilan datos provenientes de bases de datos como NHANES, acerca de hábitos alimenticios, medidas corporales y cuestionarios varios de interés.

Para proporcionar una respuesta sobre problemas genéticos y los relacionados con diferentes sintomatologías y patologías, se podrán escoger directamente distintos documentos presentes en lugares como PubMed o documentos relacionados que podremos subir directamente al modelo.

### Entrenamiento del modelo

Se deberán escoger bien los hiperparámetros como el batch-size, epochs, sequence length y el learning rate (el cual cabe la posibilidad de ajustar de manera dinámica gracias al Fast.ai).

Para entrenar los LLM se podrán usar distintos modelos extraídos de HuggingFace como RoBERTa, Mistral-7B, Llama 3-70b o gemma-7b.

## Desarrollo del prototipo de la web

### Consideraciones para el desarrollo de la web

Se tendrá en cuenta un diseño sencillo y minimalista, con un buen uso de espacios en blanco para crear una sensación de amplitud y orden. Junto con el uso de una tipografía y paletas de colores atractiva y que se adapten al estilo de la aplicación.

Por último, se podrán incluir imágenes y videos que sean relevantes para el contenido de la aplicación

Con relación a la parte de programación, se escribirá un código limpio y organizado con los comentarios pertinentes explicando su funcionamiento, junto al uso de librerías necesarias.

Para el desarrollo de la aplicación web se hará uso e Python Flask con las librerías necesarias para facilitar el desarrollo. Y en última instancia se realizarán las pruebas en distintos dispositivos y navegadores para asegurar su correcto funcionamiento.

### Pasos para el desarrollo de nuestra web

#### Identificar funcionalidades clave

Para la web, estarán presentes distintas funcionalidades:

Inicio de Sesión/Registro: Permitiremos a los usuarios crear una cuenta y/o iniciar sesión para acceder a sus datos personalizados y recomendaciones de manera segura.

Generación de Recomendaciones Personalizadas: Una vez ingresados los datos, nuestro algoritmo analizará esta información para generar recomendaciones personalizadas de estilo de vida, incluyendo dieta, ejercicio, y otros hábitos saludables.

Visualización de Resultados: Presentaremos los resultados y recomendaciones en una página de resultados de fácil lectura, que puede incluir gráficos, listas y explicaciones textuales.

Página de Contacto/Soporte: Incluiremos una sección donde los usuarios puedan enviar preguntas, comentarios o solicitar soporte técnico.

Sección de Información y Ayuda: Proporcionaremos una página con información sobre cómo utilizar la web app, la ciencia detrás de las recomendaciones y consejos generales de salud.

#### Wireframes

Procederemos a esbozar wireframes para nuestras páginas clave utilizando herramientas como Balsamiq o papel y lápiz para los bocetos iniciales. Nos centraremos en la página de inicio con opciones de inicio de sesión/registro.

En cuanto a la organización del formulario de datos genómicos y biométricos, nos aseguraremos de que el formulario esté bien organizado, agrupando los campos relacionados y utilizando indicaciones claras para cada tipo de dato.

Se incluirán elementos visuales, considerando añadir íconos o pequeñas ilustraciones que puedan hacer el formulario más intuitivo y menos monótono.

Y para facilitar la navegación, se incluirán botones claros para enviar o cargar datos, y asegurarse de que los usuarios puedan navegar fácilmente a través del formulario.

#### Diseño de Interfaz de Usuario

Con base en nuestros wireframes, diseñamos una interfaz de usuario atractiva y funcional, teniendo en cuenta las mejores prácticas de UX.

#### Prototipo interactivo

Creamos un prototipo interactivo utilizando herramientas como Figma, que nos permitirá simular la experiencia del usuario con la aplicación.

#### Selección del estilo visual

Definimos nuestra paleta de colores y tipografía para asegurar que nuestra aplicación sea visualmente coherente y atractiva. Utilicemos herramientas como Khroma para seleccionar colores que se ajusten a nuestra marca y público objetivo.

#### Recolección de Feedback

Presentemos nuestro prototipo a un grupo selecto para obtener su opinión y hagamos ajustes según sea necesario

#### Desarrollo Front-End

Con nuestro prototipo y el feedback en mano, comencemos a codificar la interfaz de usuario. Podemos usar plantillas de Bootstrap para acelerar este proceso.

#### Desarrollo Back-End con Flask

Simultáneamente, trabajamos en la configuración de Flask para nuestro back-end, asegurándonos de que los datos del usuario se manejen de manera segura y eficiente.

#### Integración Front-End y Back-End

Una vez que tengamos el front-end y el back-end listos, integremos ambos para que trabajen de forma fluida y cohesiva.

### Pruebas de validación

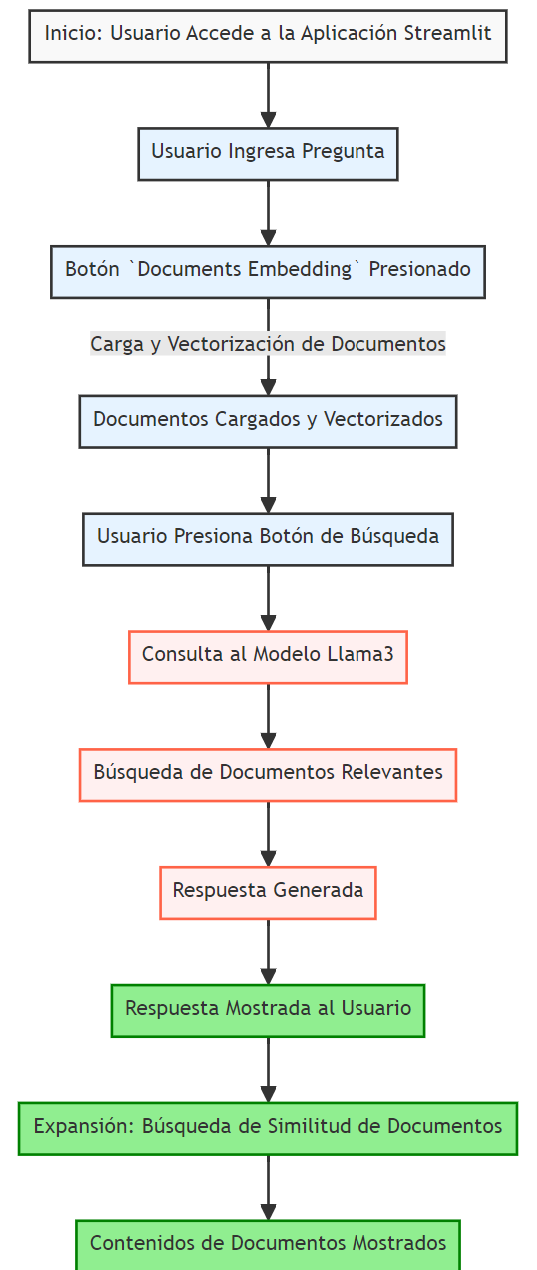
Realicemos pruebas exhaustivas para asegurarnos de que la web app funcione correctamente en diferentes dispositivos y navegadores, y que sea segura.

## Esquema y funcionamiento formal del RAG

Vamos a explicar paso a paso, como funciona el código del GitHub que hemos creado para este TFM. Y al mismo tiempo iremos revisando las herramientas que hemos utilizado, y su utilización técnica en el código Python.

### 3.3.1 Funcionamiento de llama3.py.

Mostramos esquema gráfico mediante un diagrama Mermaid ([Anexo diagramas Mermaid)](#MERMAID_LLAMA3) que ilustre el flujo de la aplicación. Este diagrama ayudará a visualizar cómo interactúan los diferentes componentes del sistema y cómo se procesan las consultas del usuario.



### 3.3.2 Integración de Tecnologías:

* **Llama 3 y Groq**: Proporcionan el backend de procesamiento de lenguaje natural y aceleración de hardware, respectivamente.
* **LangChain:** Facilita la orquestación de las diferentes etapas del proceso, desde la carga de documentos hasta la generación de respuestas.
* **FAISS:** Se utiliza para la gestión eficiente de la base de datos vectorial, permitiendo búsquedas rápidas y precisas.
* **Streamlit:** Ofrece una forma sencilla y efectiva de crear interfaces de usuario interactivas para aplicaciones de Python, permitiendo la interacción directa con el usuario.

### 3.3.3 Detalles del Código:

1. **Importaciones y Configuración Inicial:**
   * Se importan módulos necesarios como **streamlit**, **os**, y diversos componentes de **langchain**.
   * Se cargan las claves API necesarias para Llama 3 y Groq a través de variables de entorno, asegurando que el acceso a estas APIs se maneje de manera segura.
2. **Configuración de Streamlit:**
   * Se define el título de la aplicación de Streamlit, estableciendo la interfaz de usuario inicial.
3. **Inicialización del Modelo LLM y Configuración del Prompt:**
   * Se crea una instancia de **ChatGroq** usando la clave API de Groq y el modelo específico "Llama3-8b-8192".
   * Se configura un **ChatPromptTemplate** para guiar cómo el modelo debe formular respuestas, instruyéndolo a responder preguntas basadas exclusivamente en el contexto proporcionado.
4. **Vectorización y Manejo de Documentos:**
   * Se define una función **vector\_embedding** para manejar la carga, división y vectorización de documentos usando PyPDF para cargar documentos desde un directorio, seguido por FAISS para crear y manejar una base de datos vectorial de los documentos.
   * Esto prepara el sistema para realizar búsquedas eficientes de información relevante en respuesta a consultas del usuario.
5. **Interacción del Usuario:**
   * Se proporciona un campo de texto donde los usuarios pueden ingresar sus preguntas.
   * Un botón en la interfaz permite a los usuarios iniciar el proceso de vectorización de documentos, lo cual es necesario antes de que puedan realizarse búsquedas efectivas.
6. **Procesamiento de la Consulta y Presentación de Resultados:**
   * Al ingresar una pregunta y presionar el botón correspondiente, el sistema utiliza **create\_retrieval\_chain** para buscar y recuperar los documentos más relevantes, y luego **create\_stuff\_documents\_chain** para generar respuestas basadas en esos documentos.
   * Se mide y muestra el tiempo de respuesta, y las respuestas se presentan al usuario.
   * Un "expander" de Streamlit se utiliza para mostrar los resultados detallados de la búsqueda, incluyendo los contenidos de los documentos recuperados.

# Resultados

## “Título 2” del menú de estilos

### “Título 3” del menú de estilos

### “Título 3” del menú de estilos

#### "Título 4" del menú de estilos

#### "Título 4" del menú de estilos

# Discusión

## “Título 2” del menú de estilos

### “Título 3” del menú de estilos

### “Título 3” del menú de estilos

#### "Título 4" del menú de estilos

#### "Título 4" del menú de estilos

# Conclusiones

## “Título 2” del menú de estilos

### “Título 3” del menú de estilos

### “Título 3” del menú de estilos

#### "Título 4" del menú de estilos

#### "Título 4" del menú de estilos

1. Diagramas Mermaid del trabajo.

Diagrama Mermaid del **Código llama3.py**:

graph TD;

A[Inicio: Usuario Accede a la Aplicación Streamlit] --> B;

B[Usuario Ingresa Pregunta] --> C[Botón `Documents Embedding` Presionado];

C -->|Carga y Vectorización de Documentos| D[Documentos Cargados y Vectorizados];

D --> E[Usuario Presiona Botón de Búsqueda];

E --> F[Consulta al Modelo Llama3];

F --> G[Búsqueda de Documentos Relevantes];

G --> H[Respuesta Generada];

H --> I[Respuesta Mostrada al Usuario];

I --> J[Expansión: Búsqueda de Similitud de Documentos];

J --> K[Contenidos de Documentos Mostrados];

style A fill:#f9f9f9,stroke:#333,stroke-width:2px

style B fill:#e6f3ff,stroke:#333,stroke-width:2px

style C fill:#e6f3ff,stroke:#333,stroke-width:2px

style D fill:#e6f3ff,stroke:#333,stroke-width:2px

style E fill:#e6f3ff,stroke:#333,stroke-width:2px

style F fill:#fff0f0,stroke:#ff6347,stroke-width:2px

style G fill:#fff0f0,stroke:#ff6347,stroke-width:2px

style H fill:#fff0f0,stroke:#ff6347,stroke-width:2px

style I fill:#90ee90,stroke:#008000,stroke-width:2px

style J fill:#90ee90,stroke:#008000,stroke-width:2px

style K fill:#90ee90,stroke:#008000,stroke-width:2px

Referencias bibliográficas

1. Min B, Ross H, Sulem E, Veyseh APB, Nguyen TH, Sainz O, et al. Recent Advances in Natural Language Processing via Large Pre-Trained Language Models: A Survey [Internet]. arXiv; 2021 [citado 31 de marzo de 2024]. Disponible en: http://arxiv.org/abs/2111.01243

2. Egli A. ChatGPT, GPT-4, and Other Large Language Models: The Next Revolution for Clinical Microbiology? Clin Infect Dis. 11 de noviembre de 2023;77(9):1322-8.

3. Mökander J, Schuett J, Kirk HR, Floridi L. Auditing large language models: a three-layered approach. AI Ethics [Internet]. 30 de mayo de 2023 [citado 31 de marzo de 2024]; Disponible en: https://link.springer.com/10.1007/s43681-023-00289-2

4. Yu B. Evaluating Pre-Trained Language Models on Multi-Document Summarization for Literature Reviews.

5. Agapito G, Calabrese B, Guzzi PH, Cannataro M, Simeoni M, Care I, et al. DIETOS: A recommender system for adaptive diet monitoring and personalized food suggestion. En: 2016 IEEE 12th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob) [Internet]. New York, NY: IEEE; 2016 [citado 1 de abril de 2024]. p. 1-8. Disponible en: http://ieeexplore.ieee.org/document/7763190/

6. Iwendi C, Khan S, Anajemba JH, Bashir AK, Noor F. Realizing an Efficient IoMT-Assisted Patient Diet Recommendation System Through Machine Learning Model. IEEE Access. 2020;8:28462-74.

7. Yang Z, Khatibi E, Nagesh N, Abbasian M, Azimi I, Jain R, et al. ChatDiet: Empowering Personalized Nutrition-Oriented Food Recommender Chatbots through an LLM-Augmented Framework [Internet]. arXiv; 2024 [citado 1 de abril de 2024]. Disponible en: http://arxiv.org/abs/2403.00781

8. Chang J, Wang S, Ling C, Qin Z, Zhao L. Gene-associated Disease Discovery Powered by Large Language Models [Internet]. arXiv; 2024 [citado 1 de abril de 2024]. Disponible en: http://arxiv.org/abs/2401.09490