Santiago de Cali, Enero 2025.

Señores.
Pontificia Universidad Javeriana Cali.
Dr. Gerardo Mauricio Sarria.
Director de Carrera de Ingeniería de Sistemas y Computación.
Cali.

Cordial Saludo.

Por medio de la presente me permito informarle que el estudiante de Ingeniería de Sistemas y Computación, Carlos Steven Guerrero Betancourth (cod: 8955760) trabajó bajo mi dirección en el proyecto de grado titulado "Sistema accesible para la generación de dietas saludables de costo mínimo con consideraciones de economía y nutrición personalizadas" el cual se encuentra finalizado y listo para sustentación.

Atentamente,

MSc. Daniel Enrique González Gómez

Santiago de Cali, Enero 2025.

Señores.
Pontificia Universidad Javeriana Cali.
Dr. Gerardo Mauricio Sarria.
Director de Carrera de Ingeniería de Sistemas y Computación.
Cali.

Cordial Saludo.

Me permito presentar a su consideración el trabajo de grado titulado "Sistema accesible para la generación de dietas saludables de costo mínimo con consideraciones de economía y nutrición personalizadas" con el fin de cumplir con los requisitos exigidos por la Universidad para llevar a cabo el proyecto de grado y posteriormente optar al título de Ingeniero de Sistemas y Computación.

Al firmar aquí, doy fe que entiendo y conozco las directrices para la presentación de trabajos de grado de la Facultad de Ingeniería aprobadas el 26 de Noviembre de 2009, donde se establecen los plazos y normas para el desarrollo del anteproyecto y del trabajo de grado.

Atentamente,

Steven Guerrero

Carlos Steven Guerrero Betancourth
COD: 8955760

Pontificia Universidad Javeriana Cali
Facultad de Ingeniería.
Ingeniería de Sistemas y Computación.
Proyecto de Grado.

Sistema accesible para la generación de dietas saludables de costo mínimo con consideraciones de economía y nutrición personalizadas

Carlos Steven Guerrero Betancourth

Director: MSc. Daniel Enrique González Gómez

Enero 2025

Abstract

In populations living in poverty, one of the aspects that tends to be neglected is a balanced diet. This is due to various factors, among them the difficulty in planning a nutritious diet that adapts to their economic limitations. Despite efforts by the government and various entities to ensure adequate nutrition for this vulnerable population, problems related to malnutrition and nutritional imbalances continue to affect a significant percentage of the population.

One of the factors that aggravates this situation is the lack of knowledge of basic nutrition principles and the lack of accessible tools to design food plans that take into account both individual nutritional needs and budget constraints. In response to this, the proposed project proposes the creation of an accessible system for the generation of healthy diets at minimum cost with personalized economic and nutritional considerations. The proposed solution aims to provide a practical and easy-to-use tool that allows people to design diets that meet their essential nutritional needs, optimizing available resources, and promoting an improvement in their quality of life.

Keywords: nutrition, diet, health, Economically Vulnerable Population, Web Application, economy, budget.

Resumen

En las poblaciones en situación de pobreza, uno de los aspectos que tiende a quedar relegado es una alimentación balanceada. Esto debido a diversos factores, entre los cuales se encuentra la dificultad para la planeación de una dieta nutritiva que se ajuste a sus limitaciones económicas. A pesar de los esfuerzos del gobierno y de distintas entidades para garantizar una adecuada alimentación en esta población vulnerable, los problemas relacionados con la desnutrición y los desequilibrios alimenticios continúan afectando a un porcentaje significativo de la población.

Uno de los factores que agravan esta situación es el desconocimiento sobre principios básicos de nutrición y la falta de herramientas accesibles para diseñar planes alimenticios que contemplen tanto las necesidades nutricionales individuales como las restricciones presupuestarias. En respuesta a esto, el proyecto propuesto plantea la creación de un Sistema accesible para la generación de dietas saludables de costo mínimo con consideraciones de economía y nutrición personalizadas. La solución planteada tiene como objetivo ofrecer una herramienta práctica y fácil de utilizar, que permita a las personas diseñar dietas que cubran sus necesidades nutricionales esenciales, optimizando los recursos disponibles y promoviendo una mejora en su calidad de vida.

Palabras Clave: nutrición, dieta, salud, Población Económicamente Vulnerable, Aplicación Web, economía, presupuesto.

Índice general

Índice de figuras

Índice de cuadros

Capítulo 1

Introducción

Este trabajo se enmarca dentro del proyecto "Foodprice", una iniciativa de la Pontificia Universidad Javeriana Cali, el cual busca desarrollar un paquete en el lenguaje R para el cálculo de dietas de costo mínimo que garanticen una nutrición adecuada según las características de cada persona. Se seleccionó el trabajo "Sistema accesible para la generación de dietas saludables de costo mínimo con consideraciones de economía y nutrición personalizadas" como una forma de expandir el alcance del proyecto Foodprice, maximizando así su impacto en la sociedad.

La mala alimentación en Colombia representa una problemática grave que afecta, en especial, a las familias de escasos recursos [1]. Esta situación obedece a diversos factores, entre ellos la falta de educación alimentaria y nutricional, el elevado costo de los alimentos y las restricciones presupuestarias a las que estas familias se enfrentan.

Tanto el gobierno como diversas organizaciones sin ánimo de lucro han realizado esfuerzos para reducir los problemas de desnutrición [2]. Sin embargo, estos esfuerzos no han sido suficientes, ya que la problemática parece haberse agudizado en los últimos años [3]. Los programas de alimentación dirigidos a comunidades vulnerables, así como las campañas de educación nutricional, buscan contribuir a la solución de este problema, aunque aún queda mucho trabajo por hacer.

En respuesta a esta situación, se plantea el desarrollo de este proyecto, cuyo objetivo es crear un sistema que facilite el uso del software desarrollado en el marco del proyecto Foodprice, permitiendo la generación de dietas de costo mínimo que cubran las necesidades nutricionales básicas del usuario, considerando sus características específicas. Una parte fundamental del proyecto es el desarrollo de una aplicación web, con el fin de hacerlo más accesible y llegar a un mayor número de personas, al no requerir conocimientos técnicos para su uso. Con ello, se espera aportar a la reducción de los índices de desnutrición en poblaciones de bajos recursos, proporcionando una herramienta que permita identificar qué ingredientes son necesarios para una dieta lo más económica posible, incentivando así una mejor alimentación a pesar de las dificultades económicas que puedan presentarse.

Capítulo 2

Descripción del Problema

2.1. Planteamiento del Problema

La mala alimentación derivada de problemas económicos ha sido un problema constante en la historia de Colombia. [4] Esta situación afecta a diversos grupos poblacionales con necesidades nutricionales específicas que se pueden considerar grupos vulnerables, tales como niños, mujeres embarazadas, ancianos; también se tiene en cuenta que debido a la naturaleza de esta problemática, puede afectar a toda la población en general. Las afectaciones causadas por la desnutrición pueden variar dependiendo de la población específica en la que se presente, pero algunos de sus efectos negativos pueden ser: enfermedades cardiovasculares, respiratorias, óseas, entre otras.[5] El gobierno nacional cuenta con datos relevantes a este problema, tales como los valores nutricionales de alimentos que se consumen en diversas regiones del país, así como sus costes en las principales centrales de abastos, aunque no cuenta con datos precisos acerca de los precios en sitios más comunes de cara al consumidor final, tales como tiendas de barrio o supermercados de cadena, pero, a pesar de poseer todos estos datos, el gobierno no cuenta con las herramientas necesarias para su debido procesamiento para la obtención de información que permita una correcta toma de decisiones con la cual se pueda mejorar la situación de la mala alimentación en la población más pobre del país.

Como respuesta a esta problemática, se crea Foodprice, el cual es un proyecto de la Pontificia Universidad Javeriana Cali que busca desarrollar un paquete en R para el cálculo de dietas de costo mínimo que garanticen la suficiente nutrición de las personas en base a sus características, con esto se buscará analizar si es posible, a día de hoy, con los ingresos de las personas económicamente vulnerables del país, contar con una dieta saludable.

Será sobre este paquete se construirá el sistema accesible para la generación de dietas saludables de costo mínimo con consideraciones de economía y nutrición personalizadas. Esto implica varios desafíos, como la obtención de precios representativos de cara al consumidor, la ampliación del paquete Foodprice para consumirse a través de otra tecnología y conectarse de forma eficiente y escalable con aplicaciones web, y poder actualizar su base de datos de forma sencilla, manteniendo así valores que reflejen la realidad del mercado. También se requiere el desarrollo de una aplicación web que utilice este sistema, lo que implica resolver problemas relacionados con la interfaz de usuario, la accesibilidad y la optimización del rendimiento para garantizar una experiencia fluida para la población objetivo. En resumen, el problema computacional consiste en diseñar y desarrollar un sistema integral que permita expandir Foodprice de manera efectiva y accesible. Es así como se formulan las siguientes preguntas:

2.1.1. Formulación

¿Cómo desarrollar un sistema accesible para generar una dieta saludable de costo mínimo que logre un balance entre economía y nutrición según las características de la persona?

2.1.2. Sistematización

- ¿Cómo diseñar un sistema para la obtención de precios que se acoplen a la realidad del consumidor promedio?
- ¿Cómo diseñar un sistema que permita generar una dieta saludable de costo mínimo según las características de la persona y cómo diseñar una aplicación web que permita utilizar el sistema de una forma satisfactoria por parte de la población objetivo?
- ¿Cómo implementar los sistemas y la aplicación web?
- ¿Cómo validar el sistema implementado?

2.2. Objetivos

2.2.1. Objetivo General

Desarrollar un sistema que permita la generación de los costos mínimos y la asequibilidad de una dieta saludable discriminadas por edad y sexo, según su nivel de ingresos.

2.2.2. Objetivos Específicos

1. Diseñar un sistema que permita la obtención de precios que se acoplen a la realidad del consumidor promedio.
2. Diseñar un sistema que permita generar una dieta saludable de costo mínimo según las características de la persona, y una aplicación web que permita utilizar el sistema de una forma satisfactoria por parte de la población objetivo.
3. Implementar los sistemas y la aplicación web.
4. Validar el sistema implementado

2.3. Justificación

La desnutrición en Colombia es un problema que se encuentra estrechamente relacionado con la capacidad económica de las personas, siendo altamente mayor en comunidades que se encuentran en condición de pobreza o indigencia[6], causando incluso graves consecuencias como las muertes infantiles por enfermedades relacionadas con la desnutrición[7]. Esto se debe a que como resultado de la situación en la que viven estas personas, no pueden contar con una dieta balanceada que cubra

todas las necesidades calóricas y nutricionales que su cuerpo requiere. Como un posible apoyo para reducir esta problemática, se han planteado proyectos como lo es Foodprice.

Foodprice es un proyecto de la Pontificia Universidad Javeriana que busca el desarrollo de un paquete en R que permita el cálculo de dietas saludables teniendo en cuenta un presupuesto límite, sin embargo, dicho proyecto basa sus datos de precios de alimentos en el Sistema de Información de Precios y Abastecimiento del Sector Agropecuario (SIPSA) del Departamento Administrativo Nacional de Estadística (DANE), lamentablemente, dicha base de datos provee de precios basados en centrales mayoristas de las principales ciudades del país, por lo que los cálculos realizados se encuentran algo alejados de la realidad de las personas más pobres del país, además de que las dietas generadas son, de momento, para grupos específicos de población, no para individuos particulares, esto agregado a que se trata de un paquete en R, imposibilita bastante su uso por parte de las personas económicamente vulnerables para que estas puedan conocer una dieta adecuada a sus necesidades nutricionales.

Para que un proyecto así logre impactar directamente la vida de las personas que se encuentran en situaciones vulnerables, es primordial desarrollar herramientas de fácil acceso y utilización, que no requieran de un conocimiento técnico para su uso, ni de hardware especializado para su ejecución. Tomando en cuenta la situación socioeconómica de la población objetivo de este tipo de proyectos, una opción adecuada parece ser una aplicación web que funcione de forma adecuada tanto en computadoras personales como en teléfonos móviles, y que no requiera que el equipo sea de última generación para funcionar correctamente. Hoy en día, muchas personas tienen acceso a internet y pueden usar computadoras en bibliotecas públicas. Una aplicación web es sencilla, no requiere instalación ni configuración, y los navegadores están disponibles en prácticamente todos los teléfonos móviles.

2.4. Delimitaciones y Alcances

El proyecto consistirá en el desarrollo de una aplicación web que permita generar una dieta balanceada de costo mínimo, la cuál supla las necesidades nutricionales de la persona, en base a parámetros de sus características físicas.

El sistema generará una dieta de coste mínimo, de acuerdo con características físicas como lo son la edad, sexo, y condiciones particulares como el embarazo. Se contará con una base de datos que contenga la información de precios de diferentes alimentos y su valor nutricional, esta base de datos se mantendrá actualizada por medio de un sistema de web scraping. Para utilizar estos valores con el fin de generar una dieta de bajo costo adecuada a las necesidades particulares de la persona.

El proyecto contará con una base de datos que refleje la realidad de los precios que se encuentran en un supermercado colombiano, la decisión de el supermercado que se usará en el proyecto se presentará más adelante después de estudiar las opciones más viables. Los precios que se observarán

en este proyecto estarán limitados a Cali, con la posibilidad a futuro de expandirlo a más ciudades del país.

La aplicación web que se desarrolle como parte de este proyecto, tendrá como objetivo principal una fácil utilización y un funcionamiento correcto en una amplia gama de dispositivos. La información que se le entregará al usuario será una lista de alimentos crudos, junto con la cantidad mínima que necesitan consumir diariamente de cada uno, para tener una dieta que se ajuste a sus necesidades. Esto acompañado del precio para cada alimento que tendría la cantidad de este, además del precio total diario de dieta. Todos estos precios serán en peso colombiano, y las cantidades de los alimentos se presentarán en gramos, también, se entregarán gráficas como gráficos circulares, para ilustrar con mayor facilidad la cantidad de cada elemento que se consumiría diariamente en esta dieta, finalmente, se permitirá la eliminación de ciertos alimentos no preferidos por el usuario y se calculará la dieta con base en la ausencia de dicho alimento.

2.4.1. Entregables

Al finalizar el proyecto se realizará la entrega de lo siguiente:

- Documento de tesis con el diseño y la planificación del sistema de generación de dietas balanceadas de costo mínimo, así como un registro de lo que se realizó y los problemas que se hayan presentado durante el desarrollo.
- Prototipo funcional del Sistema accesible para la generación de dietas saludables de costo mínimo con consideraciones de economía y nutrición personalizadas.

Capítulo 3

Marco Teórico y Trabajos relacionados

3.1. Marco de Referencia

3.1.1. Áreas Temáticas

De acuerdo con el sistema de clasificación computacional ACM, las áreas temáticas que abarca el proyecto son:

- Applied Computing - Life and Medical Sciences - Health Informatics
- Information Systems - Decision Support Systems
- Human-Centered Computing - Accessibility

3.1.2. Marco Teórico

La malnutrición en Colombia es un fenómeno que afecta directamente a las personas más pobres de nuestro país, en el marco de este proyecto se utilizará la computación para proveer de dietas saludables de costo mínimo con consideraciones de economía y nutrición personalizada, basada en ciertos parámetros específicos. Será necesario definir algunos conceptos claves en el proyecto. Entre los cuales se encuentran: nutrición, dieta saludable, población económicamente vulnerable, aplicación web.

3.1.2.1. Nutrición

La nutrición es un aspecto crítico de la salud y el desarrollo. La buena nutrición guarda relación con la buena salud del lactante, el niño y la madre; sistemas inmunitarios más fuertes; embarazos y partos más seguros; menos riesgos de enfermedades no transmisibles (tales como diabetes y enfermedades cardiovasculares) y longevidad.[8]

3.1.2.2. Dieta Saludable

Una dieta saludable es una de las bases para la salud, el bienestar, el crecimiento óptimo y el desarrollo, y protege contra todas las formas de malnutrición. Una dieta malsana es uno de los principales riesgos para la carga mundial de morbilidad, principalmente en lo que se refiere a enfermedades no transmisibles como las enfermedades cardiovasculares, la diabetes y el cáncer.[9]

3.1.2.3. Población Económicamente Vulnerable

La línea de pobreza monetaria es el valor en dinero que necesita una persona al mes para adquirir una canasta básica de alimentos, servicios y otros bienes mínimos para vivir. Si una persona tiene un ingreso menor a este valor se considera en situación de pobreza monetaria. Por otra parte, la línea de pobreza monetaria extrema es el valor en dinero que necesita una persona mensualmente para adquirir una canasta básica alimentaria que le provea el mínimo requerimiento calórico para subsistir.[10]

3.1.2.4. Aplicación Web

Una aplicación web es un software que se ejecuta en su navegador web. Las empresas tienen que intercambiar información y prestar servicios de forma remota. Utilizan aplicaciones web para conectarse con los clientes de forma cómoda y segura. Las aplicaciones web permiten a los usuarios acceder a funciones complejas sin instalar ni configurar software.[11]

3.2. Trabajos relacionados

En esta sección se presentan diferentes trabajos de investigación, los cuales se basan en temátocas similares a las tratadas en este proyecto, o usan tecnologias relacionadas.

3.2.1. Diet recommendation system using machine learning

En este trabajo, se habla sobre la utilización de aprendizaje de máquina para generar dietas personalizadas basadas en las características de la persona, así como sus objetivos. Se utilizan métodos conocidos de aprendizaje de máquina como son random forest y LSTM, para dar recomendaciones de dietas tomando en cuenta sus características físicas para calcular su BMI y determinar si la persona tiene sobrepeso, está debajo de su peso sano, o es saludable; y en base a esto, ofrece 3 tipos de dietas, para bajar de peso, mantenerse, o aumentar. Este trabajo se relaciona con el propuesto en este proyecto, ya que en ambos se utilizará el avance tecnológico y diferentes estrategias matemáticas para determinar dietas en base a la característica de la persona, y lo que necesita para encontrarse en un estado más saludable.[12]

3.2.2. Food budget standards and dietary adequacy in low-income families

Este trabajo trata sobre el presupuesto alimentario y la adecuación dietética en familias de bajos recursos, el cual consiste en a través de diversas encuestas y análisis, estudiar los hábitos de gastos alimenticios en familias inglesas que cumplen con características económicas parecidas respecto a sus ingresos. En este trabajo se observa como hay una relación entre tanto características físicas (como sexo y edad), y características sociales (como estado civil), con el presupuesto que destinan a la alimentación, y, además, como esto influye directamente en su adecuada alimentación y posible desarrollo en los niños. Esto se relaciona con el proyecto propuesto debido a que el presupuesto

destinado para la alimentación por parte de familias de bajos recursos, es uno de los factores claves a tomar en cuenta para la generación adecuada de dietas que se adapten a sus necesidades.[13]

3.2.3. Diet Planning with machine learning

En este trabajo se trata el problema de planear dietas para niños utilizando aprendizaje de máquina para planear la alimentación de centros de cuidados de niños de Corea del Sur. Se menciona sobre cómo el aprendizaje de máquina toma en cuenta factores que con modelos matemáticos son complicadas de tomar en cuenta, como lo son la naturalidad y el disfruta de la combinación de comidas, así como los valores nutricionales que estas aportan para lograr una dieta sana y que no sea incómoda de seguir. Esto se relaciona con el trabajo del presente proyecto, ya que la capacidad de combinar comidas de formas en que la dieta suministrada no sea poco apetitosa o antinatural, son factores que se tendrán presentes también en este proyecto, además de poder cumplir con los valores nutricionales necesarios para una vida sana.[14]

Capítulo 4

Metodología, Análisis y Diseño

4.1. Metodología

4.1.1. Metodología en cascada

Para este proyecto, se decidió utilizar la metodología de **desarrollo en cascada**, la cual es una metodología de gestión de proyectos que organiza el trabajo en fases secuenciales, donde cada etapa debe completarse antes de iniciar la siguiente[15]. Se eligió esta metodología debido a que se requería definir los requisitos con exactitud desde el inicio del proyecto, además de que debido a la dificultad para reuniones constantes con el equipo del proyecto Foodprice, no era adecuado el uso de una metodología ágil.

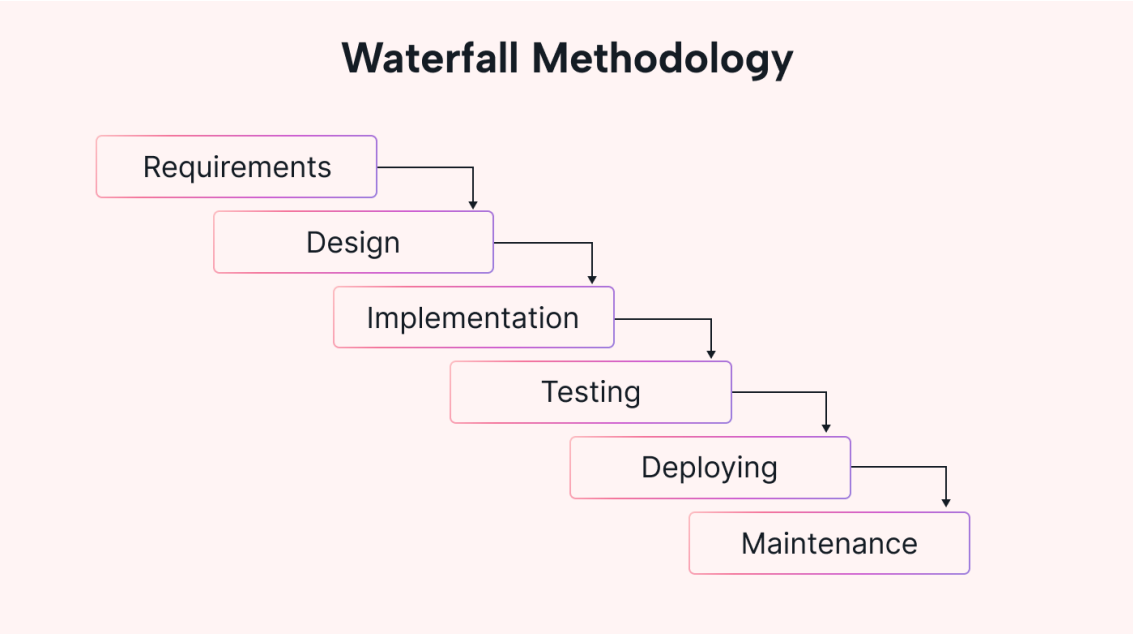

Figura 4.1: Metodología en Cascada en el Desarrollo de Sistemas

Las etapas de desarrollo en esta metodología son como se ve en la Figura 4.1 consisten en:

1. **Requerimientos:** Esta fase consiste en recolectar toda la información posible para garantizar el éxito del proyecto. Aquí se identifican las funcionalidades necesarias, las especificaciones técnicas y las expectativas del cliente. Es la fase más esencial de la metodología en cascada, ya

que al no retroceder a pasos anteriores, se requiere definir cada detalle lo mejor posible desde el principio para evitar errores en pasos posteriores.

2. **Diseño:** En esta fase se diseña cómo funcionará el sistema a nivel técnico. Se eligen las tecnologías y herramientas que se utilizarán. Y se crean diagramas como los siguientes:

   - Diseño UML de la arquitectura del sistema
   - Modelado de datos
   - Diagramas de caso de uso

3. **Implementación:** Una vez que el diseño está completo, se comienza a desarrollar el software siguiendo las especificaciones. Esta fase se centra en la codificación de las funciones definidas en las etapas anteriores.

4. **Pruebas:** Esta etapa tiene como objetivo verificar que el sistema funciona correctamente y cumple con los requisitos. Por lo cual en esta etapa el software es sometido a diversas pruebas para garantizar su correctitud. Entre las pruebas más comunes se encuentran las pruebas unitarias (de cada componente), pruebas de integración (cómo funcionan los componentes juntos) y pruebas de aceptación (ver si el sistema cumple con los requisitos del cliente).

5. **Despliegue:** En esta fase, después de que el sistema pasó todas las pruebas, se procede a desplegar el software en el entorno donde será utilizado. Y pueden empezar a interactuar los usuarios con el sistema directamente.

6. **Mantenimiento:** Después del despliegue, solo queda la fase de mantenimiento, la cual consiste en asegurar el correcto funcionamiento del sistema en un plazo de tiempo, así como realizar las mejoras y correcciones necesarias.

4.1.2. Tipo de Estudio

Con base en la finalidad de este proyecto, la cual es mejorar la calidad de vida de las personas de escasos recursos al ofrecerlos una herramienta que les provea con una dieta que se adapte a sus necesidades nutricionales y económicas; este proyecto se categoriza como un **estudio aplicado de desarrollo tecnológico, con un enfoque mixto (cualitativo - cuantitativo)**.

Esta clasificación se alinea con las características del proyecto, siendo un estudio aplicado de desarrollo tecnológico ya que busca principalmente utilizar la tecnología como un medio para crear una solución práctica y funcional que impacte positivamente la vida de sus usuarios. Y cuenta con un enfoque mixto (cualitativo - cuantitativo), ya que el impacto que genere no se medirá solo de manera numérica, sino haciendo uso también de métodos cualitativos como encuestas sobre la percepción del software y su utilidad; para así proporcionar no solo resultados medibles y replicables, sino que también contextualiza el impacto real de la solución en la vida de los usuarios.

4.2. Requerimientos

En el desarrollo de este proyecto, la fase de Análisis de requerimientos es el paso más esencial del proceso de desarrollo, debido a las razones mencionadas anteriormente de las características de la metodología en cascada. Por esto, se requirió de investigar y analizar de forma exhaustiva el panorama de los sistemas de generación de dietas, para entender cómo lograr un impacto en este campo, y alcanzar especialmente a la población objetivo, siendo estas personas en condición de pobreza o vulnerables económicamente.

A su vez, para la expansión del proyecto Foodprice, un factor clave es la obtención de precios que reflejen la realidad de lo que encontrará la persona promedio en los estantes de un supermercado.

Además de esto, otro aspecto clave del proyecto es el encontrar la mejor manera de trabajar con la aplicación en R de Foodprice, para poder utilizarla en una aplicación web de forma más sencilla y óptima, expandiendo el proyecto Foodprice, facilitando así la posibilidad de que llegue a ser usada con más frecuencia en otros proyectos.

Teniendo estos puntos en cuenta, se presenta la necesidad también de entender cuales son las mejores maneras de que una aplicación web sea funcional para la mayor variedad de dispositivos, y accesible para personas independiente de su nivel de experticia con la tecnología.

Para poder lograr una fase de análisis de requerimientos exitosa, se decidió comenzar por definir requerimientos de investigación, funcionales, y no funcionales. Después de esto, se procederá al desarrollo de los requerimientos de investigación, la definición del arquetipo de usuario (población objetivo), y diagrama de casos de uso. Después de esto se estudiará la decisión de las herramientas que se usarán, junto con sus ventajas y desventajas. Y con esto se procederá a continuar con la fase de diseño teniendo unas bases sólidas para un proyecto bien estructurado.

4.2.1. Levantamiento de Requerimientos

Es preciso que se aclare que los temas de investigación a pesar de que se plantearon en el proyecto, no forman parte de los requerimientos de un desarrollo de software por ello se manejan como un tipo de requerimiento distinto a los usuales: Funcional y No funcionales.

Como Requerimientos Funcionales se entiende que son todos aquellos requerimientos que el usuario final exige específicamente como prestaciones básicas que debe ofrecer el sistema. Todas estas funcionalidades deben estar necesariamente incorporadas al sistema como parte del contrato. Estos se representan o establecen en forma de entrada que se le dará al sistema, la operación realizada y la salida esperada. Son básicamente los requisitos declarados por el usuario que se pueden ver directamente en el producto final, a diferencia de los requisitos no funcionales[16].

Como Requerimientos no funcionales se entiende que son todos aquellos requerimientos que tienen que ver con la calidad, son básicamente las restricciones de calidad que el sistema debe satisfacer según el contrato del proyecto. La prioridad o la medida en que se implementan estos factores varía de un proyecto a otro. También se les llama requisitos no conductuales[16].

4.2.2. Requerimientos de investigación

Cuadro 4.1: Tabla requerimientos de investigación

RI# - Título	**Descripción**	**Criterio de aceptación**
RI01 – Características de sistemas de generación de dietas	▪ Se debe lograr entender cuales son las características principales de los sistemas de generación de dietas que existen en la actualidad. ▪ Observar que opciones permiten estos sistemas, y que limitantes tienen	▪ Se investiga sobre software que haga un trabajo relacionado, idealmente que genere dietas. ▪ Se enumeran características que sean clave para la generación de dietas en estos software. ▪ Se determina que características y opciones tendrá el software de este proyecto.

RI02 – Mejor método de obtención de datos con impacto real	▪ Se estudia que supermercados reflejan mejor la realidad del consumidor promedio, y por qué motivos. ▪ Se determina que supermercado es pertinente para este proyecto. ▪ Se explican las razones del por qué se elije este. ▪ Se analiza de que maneras se pueden obtener los datos que reflejen los precios reales.	▪ Se comparan de forma eficaz los supermercados con mayor impacto de colombia. ▪ se proveen razones lógicas que soporten la decisión del mercado que se usará para este proyecto. ▪ Se establece como se obtendrán los datos de dicho supermercado.

4.2.3. Requerimientos Funcionales

Cuadro 4.2: Tabla requerimientos funcionales

RF# - Título	**Descripción**	**Criterio de aceptación**

RF01 – Generación de las dietas	▪ El sistema debe permitir al usuario generar una dieta diaria que cubra sus necesidades nutricionales básicas al menor costo posible, basado en la información de precios de alimentos.	▪ La dieta generada debe cubrir al menos el 100 % de los requerimientos diarios de calorÃas. ▪ La dieta generada debe ser de costo mínimo. ▪ La dieta generada debe tomar en cuenta las características que el usuario elija, así como las restricciones. ▪ La dieta debe indicar el valor calórico o nutricional diario que suple, así como el costo total de comprarla en el mercado del que se obtenga la información ▪ Cada alimento que hace parte de la dieta, debe ir acompañado de una imagen, precio, información nutricional, y link sobre donde pueden encontrar el producto.
RF02 – Visualización de las dietas	▪ Se debe poder visualizar de forma sencilla y clara la dieta que genere el sistema.	▪ El sistema debe mostrar de una forma clara la dieta generada para dicho usuario en base a las características que este definió.

RF03 – Selección del tipo de dieta	▪ El sistema debe permitir elegir entre 3 tipos de dietas, las cuales pueden suplir necesidades calóricas (cubre la cantidad de calorías necesarias para el funcionamiento del cuerpo), nutricionales (cubre la cantidad de calorias y nutrientes para el funcionamiento del cuerpo), o ser una dieta recomendada o saludable, y variada.	▪ El sistema debe permitir al usuario seleccionar cual de estas 3 dietas prefiere generar. ▪ El sistema debe adaptarse a la dieta generada, y generar la correspondiente.
RF04 – Selección de características del usuario	▪ El sistema debe permitir seleccionar una serie de caracteísticas, las cuales tienen un impacto en el tipo de dieta necesaria, y adecuar la dieta a estas.	▪ El sistema debe contar con una lista de características físicas, las cuales tengan un impacto en la dieta que se genere. ▪ El sistema debe tener en cuenta estas características a la hora de generar la dieta para el usuario.

RF05 – Selección de restricciones	▪ El sistema debe permitir seleccionar una serie de alimentos los cuales no desea que hagan parte de la dieta, y estos deben ser tomados en cuenta y no hacer parte de la dieta final, siendo reemplazados por otros alimentos.	▪ El sistema debe contar con una lista de alimentos, los cuales pueda seleccionar el usuario para indicar que no desea que hagan parte de su dieta. ▪ Debe permitir seleccionar varios alimentos, y todos tenerlos en cuenta. ▪ La dieta generada no debe tener ninguno de los alimentos que se restringieron.
RF06 – Nombre del usuario de la dieta	▪ El sistema debe permitir al usuario ingresar su nombre, y este dato debe verse reflejado en su dieta, esto para permitir con mayor facilidad al usuario saber que esa dieta en específico le pertenece.	▪ El sistema debe contar con un campo en el que el usuario pueda escribir su nombre. ▪ El nombre del usuario debe verse en la dieta.

4.2.4. Requerimientos No Funcionales

Cuadro 4.3: Tabla requerimientos no funcionales

RNF# - Título	Descripción	Criterio de aceptación

RNF01 – Accesibilidad de la página - colores	▪ El sistema debe cumplir con los estándares de accesibilidad respecto a contrastes de colores definidos en la Web Content Accessibility Guidelines (WCAG)	▪ El contraste entre el fondo y los textos debe tener un ratio de contraste de 4.5:1 para texto normal (pequeño), y 3:1 para texto grande (18 pt o más).
RNF02 – Tiempos de carga razonables	▪ Se debe asegurar que los tiempos de carga al usar la herramienta sean razonables. Sobre todo al ir enfocado su uso a una población que podría no tener un internet tan veloz.	▪ El sistema no debe tardar más de 3 segundos en la carga inicial de la página. ▪ El sistema no debe tardar más de 5 segundos en la carga del plan de dieta. ▪ Debe haber un tipo de indicador de que la página se encuentra cargando, para así no haya problema aunque la carga tome más tiempo por factores externos.

RNF03 – No hay errores con la combinación de características y restricciones	▪ Se debe poder asegurar que sin importar que caracteísticas y restricciones se mezclen en la aplicación, esta siga funcionando con normalidad.	▪ Asegurar que ninguna mezcla de características y restricciones genera un error en el sistema. ▪ Asegurar que se genere al menos una dieta de costo mínimo sin importar la mezcla de características y restricciones que se tenga.
RNF04 – No permitir que se restrinjan todos los alimentos	▪ Debido a que no se puede generar una dieta sin alimentos, es necesario asegurar que el sistema no permita que el usuario restrinja todos los alimentos.	▪ El sistema debe indicarle al usuario que debe seleccionar al menos un alimento. ▪ El sistema no debe permitir que el usuario restrinja todos los alimentos.
RNF05 – Si los parámetros de entrada permanecen constantes, la dieta generada debe ser consistente y presentar los mismos resultados	▪ Si los parámetros de características y restricciones son iguales, el sistema debería arrojar la misma dieta, esto mientras no se hayan presentado cambios en la base de datos de alimentos.	▪ En el caso de generar una dieta en diferentes ocasiones, con las mismas entradas en características y en restricciones, debe en todos los intentos generar la misma dieta. Mientras la base de datos no haya cambiado.

RNF06 – Conectar el paquete Foodprice a través de Python a la aplicación web	▪ El sistema debe funcionar en una constante comunicación entre R y Python, donde el middleware en Python se encargue de pasarle la información al paquete Foodprice que necesita, y el paquete entregue la dieta de costo mínimo relacionada con la información ingresada.	▪ El sistema debe ser capaz de comunicarse con el paquete de R de alguna forma a través de Python ▪ El sistema no debería interactuar con el sistema de R de forma directa sin Python. ▪ La conexión entre R y Python debe garantizar el correcto funcionamiento del proyecto,
RNF07 – Testear la aplicacioń web con usuarios	▪ El sistema debe ser probado y funcionar adecuadamente con usuarios.	▪ El sistema debe ser probado con usuarios antes de su despliegue final. ▪ El sistema debe pasar dichas pruebas para considerarse que está en un estado aceptable. ▪ Se deben documentar las opciones de mejora que los usuarios sugieran para trabajos futuros.

RNF08 – Pruebas de rendimiento de la aplicacion	▪ El sistema debe ser evaluado mediante herramientas de automatización especializadas en pruebas de rendimiento.	▪ Antes del despliegue final, se deben realizar pruebas automatizadas para garantizar el cumplimiento de los requisitos de rendimiento. ▪ El sistema debe pasar dichas pruebas para considerarse que está en un estado aceptable. ▪ Se debe simular un entorno de alta carga y estrés para analizar el comportamiento del sistema bajo estas condiciones. ▪ Los resultados de las pruebas de estrés deben ser documentados detalladamente, incluyendo métricas clave como tiempos de respuesta.

4.2.5. Desarrollo de requerimientos de investigación

Con base en los requerimientos de investigación que se definieron anteriormente, se procederá a su desarrollo, para cumplir con sus criterios de aceptación.

4.2.5.1. RI01 – Características de sistemas de generación de dietas

Para poder realizar de forma satisfatoria este proyecto, es necesario entender las bases que existen en la actualidad respecto a sistemas que cumplan un rol similar. Además de aquellos que se observaron en la sección 3.2, podemos encontrar otros sistemas como lo son una herramienta del proyecto PlaSA Colombia, el cual se define como un "espacio de fácil acceso en el que cualquier persona interesada en el universo del sistema alimentario encontrará desde datos hasta narrativas a partir de información veraz, sencilla y actualizada" [17]; este proyecto cuenta con una herramienta que muestra tres opciones de dieta, una saludable, una nutritiva, y una de subsistencia; y muestra el

precio y los alimentos en los que se encuentra distribuida una dieta de costo mínimo para estos tipos de dietas. Además de esto, cuenta con una gráfica que muestra el costo diario de la dieta para diferentes grupos de edades, tanto para hombres como para mujeres.

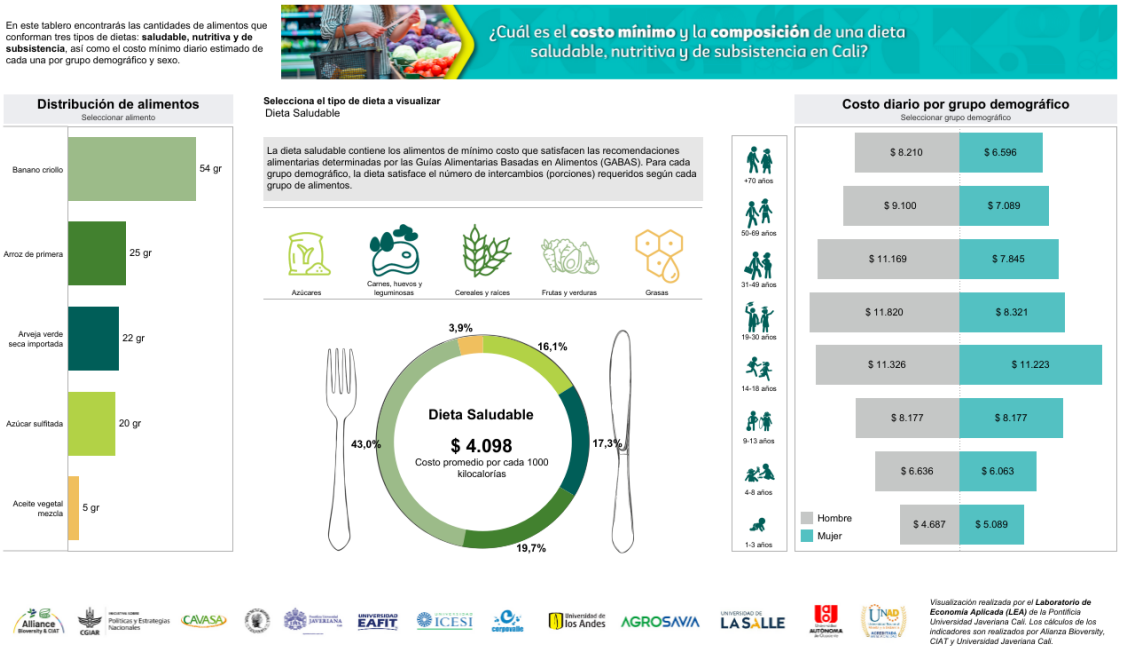

Figura 4.2: ¿Cuál es el costo mínimo y la composición de una dieta saludable, nutritiva y de subsistencia en Cali?

Este proyecto, como se puede observar en la figura 4.2, presenta información útil, de una forma organizada, sin embargo, presenta algunos problemas.

- La base de datos de la que sacan los precios de sus alimentos, se encuentra bastante desactualizada, siendo esta del 2022.
- La base de datos toma los precios del Sistema de Información de Precios y Abastecimiento del Sector Agropecuario (SIPSA) , por lo que estos no reflejan los precios con los que se encontrará el consumidor final en las estanterías.
- No funciona de forma correcta la opción de seleccionar alguno de los Costo diario por grupo demográfico; al seleccionar alguno el precio cambia, pero no refleja los datos que debería en la grafica de distribución de alimentos o en la gráfica central.
- No toma en cuenta la posibilidad de que una persona no pueda o no desee consumir un alimento en particular, por lo que puede que la dieta que provea no sea útil para el usuario

A pesar de estos problemas, el acercamiento de PlaSA Colombia al reto que presenta crear un sistema así es un gran avance, aportando en el panorama de herramientas útiles para disminuir los

problemas de nutrición en las poblaciones economicamente vulnerables. Se destaca que cuenta con características como:

- Gráficas que muestran de forma visual los datos.
- Gráfica de la distribución de cada alimento en el total de la dieta diaria.
- Costo para dietas de diferentes grupos de edades y sexo.
- Una interfaz organizada y llamativa, con colores que ayudan a visualizar y diferencias los diferentes componentes de la dieta

Otro ejemplo de un sistema de generación de dietas, aunque con un enfoque mucho más general y comercial, sería Diet Creator, el cual es un software para uso de nutricionistas, que les ayuda a crear dietas de forma rápida y sencilla para los usuarios de dichos nutricionistas, así como les ayuda a llevar un control de los datos, para de esta forma tener un mejor control del avance y el manejo de dichas dietas.

Diet Creator, a pesar de no tratar un tema totalmente alineado con este proyecto, cuenta con algunas características de las que se puede aprender, así como un entendimiento de las dificultades y la importancia de utilizar una base de datos actualizada en un sistema como este, tal como se evidencia en la cita "Todo software de nutrición debe tener una base de datos de alimentos de fuentes científicas actualizadas, también debería incorporar los productos reales del mercado, pero esto entraña dificultades debido a que los productos son tan cambiantes que armonizar la información es todo un reto global."[18].

Este sistema es mucho más comercial, ya que no es gratuito, y uno de sus principales puntos de venta es ayudar a nutricionistas en su trabajo. Sin embargo, es un software muy completo, que cuenta con una aplicación muy intuitiva y con muchas herramientas, pero para lo relacionado con este proyecto, se pueden mencionar las siguientes características:

- Permite generar dietas personalizadas, con diferentes propósitos.
- Permite seleccionar alimentos que no se quieran incluir en la dieta.
- Toma en cuenta muchos datos sobre el usuario, para que la dieta se adapte lo mejor posible a este.
- Provee toda la información de una tabla nutricional para cada producto
- Entrega la información de forma sencilla y profesional
- Permite descargar la información para su impresión o uso en general.

Sin embardo, a pesar de lo útil que es esta herramienta, no cubre todas las necesidades que requiere este proyecto, al no incluir cosas como:

- Precios para los alimentos
- Opción de dieta de costo mínimo

Con lo observado en estos sistemas, se eligen entonces como caracteísticas clave para el sistema de este proyecto las siguientes.

- Contar con una lista de alimentos variados, con los cuales se puedan crear diversas dietas en caso de que se restrinjan alimentos claves.
- Contar con la información nutricional asociada a cada alimento.
- Contar con una buena experiencia de usuario, que no oculte las funcionalidades; al contrario, facilite el uso de la herramienta a su máxima capacidad.
- Contar con precios que reflejen la realidad del mercado.
- Contar con las características físicas: edad, sexo, y estado de embarazo. Para así cubrir los factores que más pueden afectar las necesidades nutricionales de una persona.
- Contar con 3 variedades de dietas mínimas, para adaptarse a las necesidades del usuario, siendo estas una dieta adecuada en calorias, adecuada en nutrientes, y una dieta saludable.

4.2.5.2. RI02 – Mejor método de obtención de datos con impacto real

Tal como se vio en la sección anterior, uno de los factores más complicados a la hora de crear un sistema como este, es contar con datos que reflejen la realidad del consumidos final, que sean al menos muy similares a aquello que encontrará una persona al dirigirse a un supermercado.
Para resolver esta problemática, este poryecto hará uso de una base de datos, que se obtendrá haciendo uso de técnicas de web scraping, el cual es el proceso de recopilar contenidos de acceso pÃºblico de un sitio web y guardarlos en una base de datos, un archivo o una hoja de cÃ¡lculo para su posterior análisis[19]. Esto se requiere debido a que los supermercados grandes de Colombia no cuentan con apis de uso público con las que se pueda acceder a sus datos.

Teniendo claro que metodología se usará para conseguir los datos, se presenta entonces la cuestión de definir cual es el supermercado que mejor represente la realidad de las personas de Cali, y por qué motivos.

Al observar datos del artículo “Los 40 supermercados más grandes de Colombia”[20]. Se puede observar que los 3 supermercados lideres del sector en el 2024 fueron Tiendas D1, Tiendas ARA, y Almacenes Éxito. Por lo que se decidió que se trabajaría con una de estas tres para este proyecto.

Sin embargo, a pesar de que Tiendas D1 y ARA son las primeras opciones, se debe tomar en consideración que ambas pertenecen al modelo “Hard discount”[21], en el cual uno de los factores

emblemáticos de este modelo, es un portafolio de productos limitados, en el que priman las marcas propias. Por lo que a pesar de presentar en general precios más bajos. Su baja variedad de productos hace que no sean los candidatos idóneos para un proyecto como este, ya que limitarian en gran cantidad la variedad de dietas que se puedan generar.

Con esto a favor, se decide optar por trabajar con Almacenes Éxito para el propósito de este proyecto, los cuales cuentan con 522 tiendas a nivel de Colombia[22], de los cuales 9 están ubicados en la ciudad de Cali, esparcidos a través de la ciudad, cubriendo así la mayoria de zonas de la localidad.

4.2.6. Arquetipo de Usuario

Este proyecto apunta a proporcionar una herramienta para mejorar la calidad de vida de personas en situación de pobreza monetaria o económicamente vulnerables. Por lo que el usuario al que apunta este proyecto son personas en dichas condiciones, y según los resultados entregados por el DANE, en el periodo 2021 - 2023 [23], el rango de edad de la población con mayor porcentaje en condición de pobreza en Colombia es entre los 25 y 35 años, con un 40,8 %, sin embargo, las poblaciones en edades hasta los 25 años, al igual que entre los 35 y 45 años presentan un personaje muy cercano a esta, por lo que se propone como edad para el arquetipo de usuario desde los 18 hasta los 45 años; aclarando que cualquier persona en cualquier rango de edad puede hacer uso de este sistema, ya que soporta todas las edades para la generación de dietas.

Como se puede ver en la figura 4.3, otro factor importante a tener en cuenta a la hora de pensar en el arquetipo de usuario es el nivel de escolaridad, en el cual se muestra que tomando en cuenta este perfil, la población con nivel educatico Ninguno o Primaria es la que tiene un porcentaje de personas en condición de pobreza monetaria mayor, por lo que será la que se tendrá más encuenta para este proyecto.

Perfil del jefe de hogar		Nacional	Cabeceras	Centros poblados y rural disperso	13 ciudades y A.M.	Otras cabeceras
Sexo	Hombre	29,5*	26,4*	38,3*	23,0*	31,1*
	Mujer	37,7*	35,7*	46,1*	30,4*	42,5*
Edad	Hasta los 25 años	39,5*	38,2*	42,3*	34,6*	42,3*
	Entre 26 y 35 años	40,8*	38,9*	47,1*	34,2*	45,1*
	Entre 36 y 45 años	37,7*	35,3*	45,6*	31,4*	40,5*
	Entre 46 y 55 años	29,6*	27,1*	38,1*	23,1*	32,4*
	Entre 56 y 65 años	25,5*	23,5*	32,8*	19,9*	29,2*
	Mayor a 65 años	26,0*	22,7*	37,7*	18,1*	29,3*
Nivel Educativo	Ninguno o primaria	43,0*	41,5*	45,2*	37,2*	45,4*
	Secundaria	36,9*	36,6*	38,1*	33,9*	40,0*
	Técnica o tecnológica	22,4*	22,1*	25,7*	19,7*	26,1*
	Universidad o posgrado	7,3*	7,1*	13,3*	6,1*	9,4*
Situación laboral	Ocupados	29,6*	27,3*	37,0*	24,1*	31,9*
	Desocupados	60,2*	60,2*	60,5*	55,9*	65,3*
	Población fuera de la fuerza de trabajo	37,5*	33,6*	50,5*	27,0*	42,0*
Posición Ocupacional	Asalariados	18,6*	19,0*	16,9*	18,1*	20,4*
	Patronos y Cuenta Propia	40,4*	37,1*	47,9*	32,2*	42,9*
Seguridad social (Pensiones)	Afiliado	11,2*	11,7*	6,3*	12,5*	10,1*
	No afiliado	43,7*	43,9*	43,3*	41,1	46,6*
Total		33,0*	30,6*	41,2*	26,4*	36,4*

Figura 4.3: Incidencias de pobreza monetaria según perfil del jefe de hogar (porcentaje)

- Edad: 18 - 45 años
- Nivel educativo: Ninguno o primaria
- Idioma nativo: Español
- Intención: tener una opción de dieta de costo mínimo que se adapte a sus necesidades calóricas o nutricionales.

4.2.7. Diagrama de casos de uso

Con base en los requerimientos anteriores, se plantea el diagrama de casos de uso en la figura 4.4, en el cual se observan los diferentes casos de uso y como se extienden o incluyen otros casos de uso. Como se puede observar en el diagrama, los casos de uso para el usuario se centran en interactuar con el sistema para adaptar la dieta a sus características y restricciones, y visualizar la dieta.

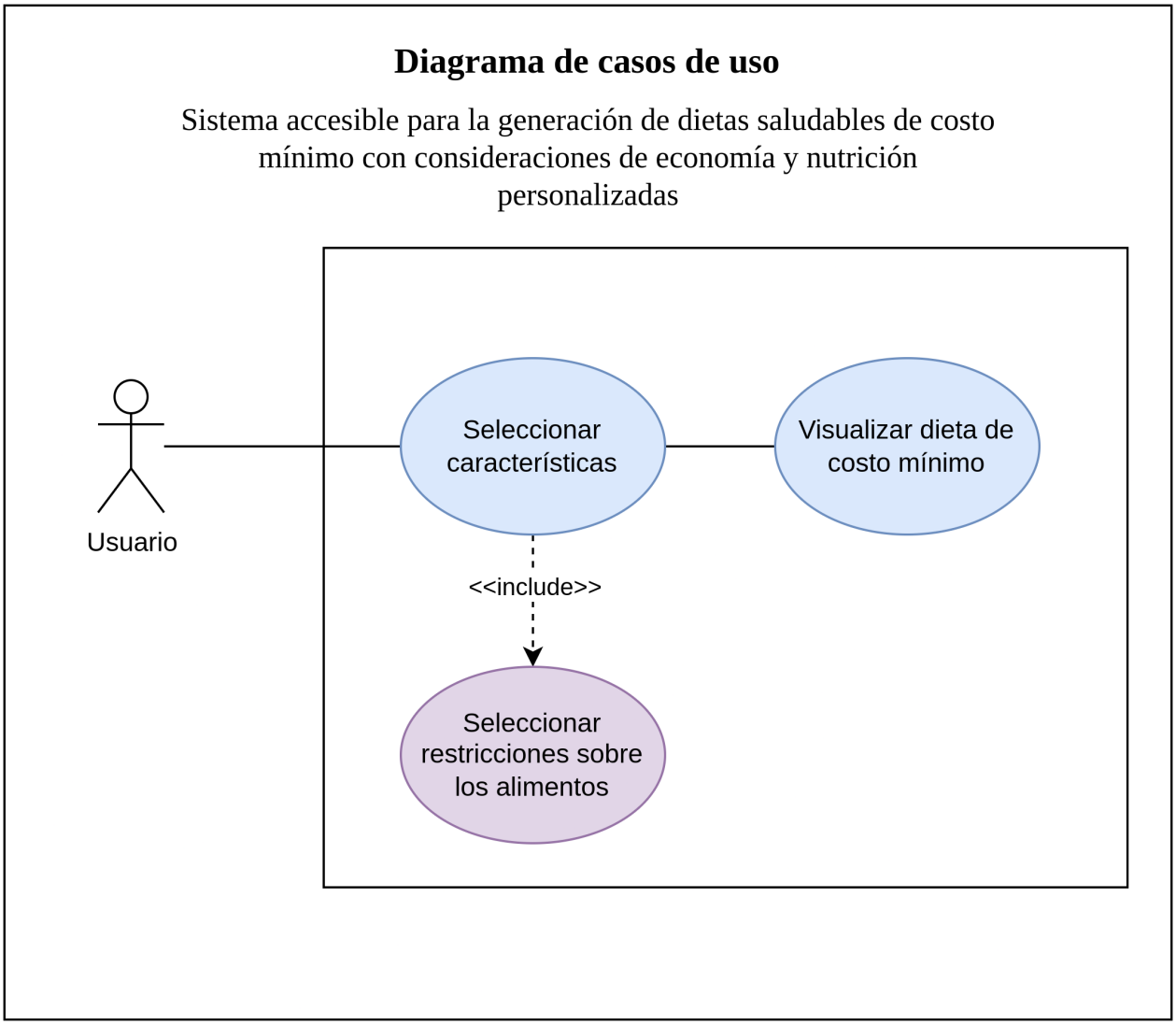

Figura 4.4: Diagrama de casos de uso

4.2.8. Decisiones de tecnologías

Al ser este un proyecto de ingeniería, las tecnologías que harán parte del proyecto son uno de los factores centrales para el desarrollo de este. Por lo que se debe tener claridad desde el principio de cuales tecnologías conformarán este sistema.

4.2.8.1. R

Como base de este proyecto, se encuentra el proyecto Foodprice, desarrollado por la Pontificia Universidad Javeriana Cali en colaboración con Alliance Bioversity & CIAT; el cual es un paquete en R, el cual es un entorno y lenguaje de programación con un enfoque matemático[24].

Figura 4.5: R logo

Este paquete permite a través de un modelo matemático calcular dietas de costo mínimo en base a algunos parámetros y restricciones que elija el usuario; sin embargo, este paquete cuenta con muchos inconvenientes para su uso. Principalmente, R no es un lenguaje amigable para el público general, siendo considerado un lenguaje duro de aprender y masterizar[25]. Sumado a esto, el paquete Foodprice no está desarrollado siguiendo las mejores prácticas y estándares de programación, por lo que su modificación y uso puede llegar a complicarse aún más. Agregando además el hecho de que R es apenas el 21 lenguaje más usado por programadores a nivel mundial, con solo un 4.3 % [26], termina siendo este un lenguaje que a pesar de su poder para cálculos estadísticos y ciencia de datos, dificulta mucho su uso generalizado; y sobre todo practicamente imposibilita el uso por la población a la que quiere impactar, siendo esta personas en situacion de pobreza, que probablemente no cuenten con los conocimientos y habilidades para poder utilizar un software de estas características de forma correcta y clara.

Figura 4.6: logo proyecto Foodprice

Por todos estos motivos, se decide entonces que un factor base para este proyecto es crear un middleware que permita una conexión más sencilla y natural con el paquete de Foodprice, además de estar construido en una tecnología más popular, para así facilitar la posibilidad de que otras entidades o personas puedan consumir o expandir el proyecto Foodprice, llegando así a más personas, potencialmente mejorando la calidad de vida de personas que requieran de una dieta de costo mínimo que se adapte a sus necesidades. Además de esto, se presenta la necesidad de utilizar el paquete Foodprice como una API para poder comunicarse de forma sencilla con python, para lo cual se usará Plumber.

4.2.8.2. Plumber

Debido a la complejidad de trabajar directamente código de R en python, y que las opciones que existen pueden presentar complejidades, sobre todo al trabajar en windows, se optó por usar Plumber para transformar el paquete Foodprice en una API, la cual se consumirá mutuamente con otra API de Python, para así comunicarse en lo que se necesite, e invocar las funcionalidades de R desde Python, a través de su API.

Figura 4.7: logo plumber

Como se mencionó anteriormente, Plumber es un paquete en R que permite crear una web API con solamente decorar el codigo de R que ya se tiene, permitiendo así una fácil conexión que no requiere de modificar fuertemente el programa ya creado[27].

4.2.8.3. Python

Como respuesta a lo mencionado anteriormente, se propone Python como lenguaje para el desarrollo del middleware que usará a Foodprice. Esta decisión se toma debido a varios factores, entre los principales la popularidad y el extenso uso que tiene Python en la comunidad de desarrolladores, siendo el tercer lenguaje más usado en 2024 a nivel mundial, con un 51 % de uso[26]. Estos factores abrirán las puertas al uso del paquete Foodprice para muchos desarrolladores, en conjunto con el middleware en Python.

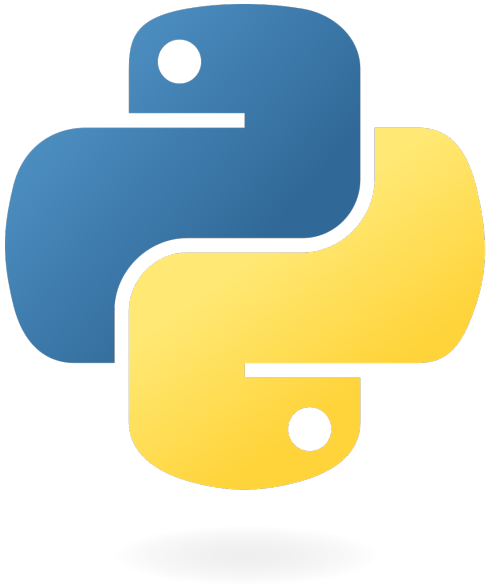

Figura 4.8: logo Python

Además de esto, Python permite una conexión más sencilla y eficiente con aplicaciones web, por lo que es ideal para el propósito de este proyecto.

4.2.9. FastAPI

Para conectar Python como una web API, se hace uso de FastAPI, un web framework rápido y moderno, el cual permite crear Apis con Python centrándose en la velocidad de rendimiento, y velocidad de codificación al no ser compleja de usar[28].

Figura 4.9: logo FastAPI

Al desarrollar ya ambas APIs, estas se conectarán entre si, para comunicarse, y pasar la información que se requiera, siendo la aplicación de Python la que consuma realmente a la de Foodprice en R, y después de eso se encargue de pasarle la respectiva información al front-end para visualizar todo en la aplicación web.

4.2.9.1. ZOD

Zod es una biblioteca de validación y análisis de esquemas de datos diseñada para aplicaciones de JavaScript y TypeScript[29]. Se utiliza en este proyecto para encargarse de facilitar la definición, validación y transformación de datos de manera estricta y segura.

Figura 4.10: logo Zod

4.2.9.2. Next.js

Next.js es un framework flexible de React que te ofrece bloques de construcción para crear aplicaciones web rápidas y completas[30]. Este será el framework que se utilice en la parte de la construcción de la aplicación web, siendo este el en cargado del front-end, conectándose con el back-end que serán Python y Foodprice. Aprovechando la capacidad de Next.js para crear una aplicación con elementos visuales interesantes, accesible y de uso intuitivo.

Figura 4.11: logo Next.js

4.2.9.3. React

React es una biblioteca de JavaScript de código abierto desarrollada por Meta (anteriormente Facebook), diseñada para construir interfaces de usuario (UI) interactivas y dinámicas de forma eficiente[31]. Al ser Next.js un framework de React, se utiliza este para el proyecto, aprovechando su facilidad para trabajar de forma modular con componentes para crear así una buena interfaz de usuario.

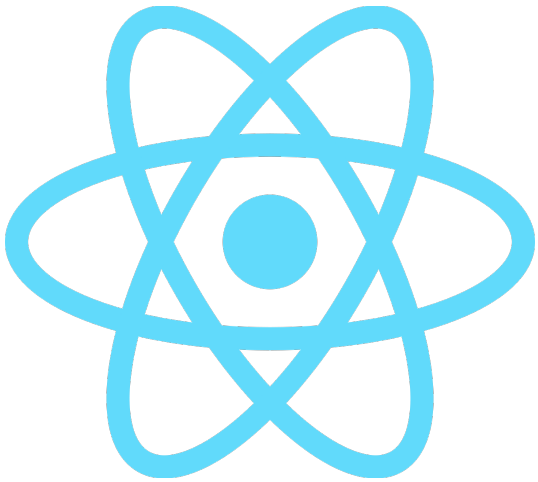

Figura 4.12: logo React

4.2.9.4. React hook form

React Hook Form es una biblioteca ligera y eficiente para manejar formularios en aplicaciones React. Está diseñada para simplificar la validación y el manejo de datos de los formularios, aprovechando las características de los React Hooks para optimizar el rendimiento y proporcionar una experiencia de desarrollo más fluida[32].

Figura 4.13: logo React hook form

4.2.9.5. Selenium

Para conseguir la información de los alimentos que representen la realidad de lo que se ve en un supermercado de Cali, se llegó a la conclusión que se requiere de hacer web scraping, ya que estos datos no tienen una API pública para poder consumirlos. A pesar de que el web scraping es una práctica moralmente ambigüa, los datos que se conseguirán en este proyecto no se utilizarán para generar ninguna ganancia monetaria, y por el contrario se usarán para crear un bien para la sociedad. Por lo que con base en que no hay una legislación clara al respecto, y lo que en casos generales se considera un buen uso de esta técnica que no tiene problemas legales [33], el uso en este proyecto no debería presentar algún problema.

Figura 4.14: logo Selenium

Con esto claro, se utilizará Selenium como herramienta de automatización con la que se obtendrán los datos. Todos los datos que se obtienen a través de Selenium para este proyecto son públicos, solo se están extrayendo, y agrupando en una estructura definida, automatizando la búsqueda en el portal web de Almacenes Éxito, y tomando la información relevante para este proyecto del producto específico.

4.2.9.6. Dataframe

El dataframe no es exactamente una tecnología en sí, pero es la manera en la que los datos que son scrapeados con Selenium serán guardados. Con el fin de poder acceder a ellos en cualquier momento desde el paquete en R.

4.2.9.7. MongoDB

Debido a que no se pueden realizar queries eficientes al dataframe por ID, se deció que era necesario contar con una base de datos desde la cual se pueda obtener la lista de alimentos, se seleccionó la base de datos MongoDB debido a que el modelo de datos que se maneja es simple y no cuenta con relaciones, además de que los datos de alimentos son guardados en formato JSON, que es el formato por defecto de MongoDB.

Figura 4.15: logo MongoDB

4.3. Diseño

En este apartado se presenta el diseño de la arquitectura de la aplicación, el modelo, así como un bosquejo sencillo inicial de la interfaz de la aplicación. Para esto, se presentan los diagramas realizados para describir el flujo de la aplicación, desde aspectos como el usuario dentro de la aplicación, hasta el funcionamiento de esta.

4.3.1. Arquitectura

La arquitectura es un eje central en todo proyecto de software. En este caso, debido a como cada parte del proyecto se integra entre si, este sistema pertence a la **Arquitectura Modelo - Vista - Controlador**, esto debido a que se cuenta con tres ejes principales:

- Paquete Foodprice: desarrollado en R, este cumple el papel de **modelo**, encargándose de toda la lógica de consultar a la base de datos, hacer los cálculos respectivos con base en la información que recibió del modelo, y entregarlo al modelo para que sea visualizado después por la vista.

- Middleware Python: Aquí se maneja la interacción entre la vista y el modelo, recibiendo este la información de la vista, haciendo pequeñas tareas de transformar datos para que el modelo los reciba correctamente, e invocando al modelo para que se encargue de toda la lógica y consultas a base de datos (en este caso dataframe); de esta forma cumpliendo el rol de **controlador**. Sin embargo, en este proyecto no se puede considerar un controlador puro, ya que realiza algunas funciones un poco más complejas, como traer datos de la base de datos de MongoDB para poder tener una eficiencia más alta, ya que el dataframe con el que se trabajó en R no permite hacer queries en base a un id, y el proceso de relacionar cada alimento para traer toda su información sería muy lento. Aún así se considera que ya que en su mayoría cumple con la tarea de un controlador, que puede aplicarse este rol.

- Front-end: El front-end desarrollado en Next.js, el cual se encarga únicamente de lo relacionado con la **vista** y la interacción con el usuario.

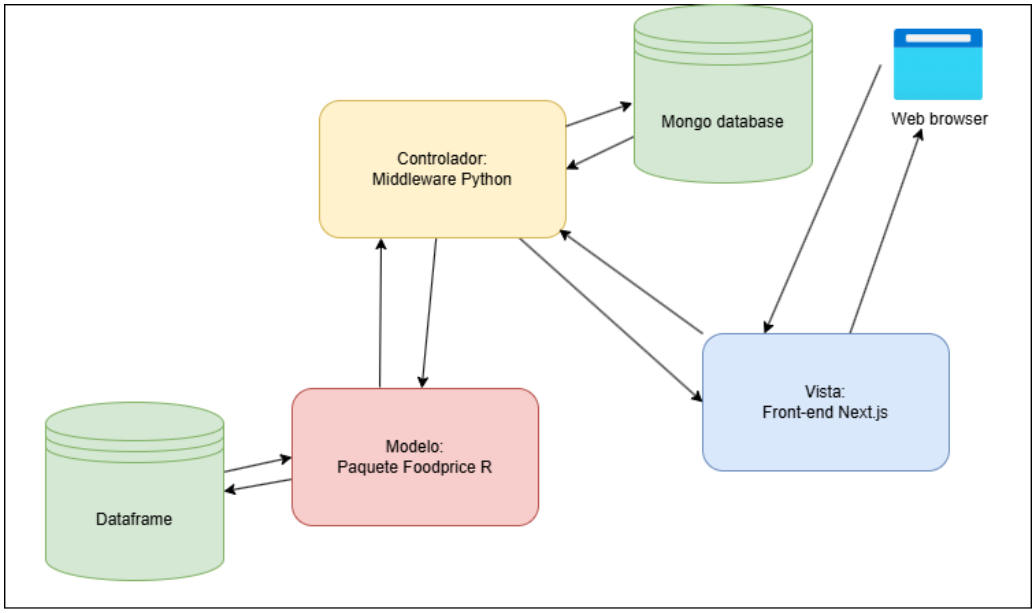

Figura 4.16: Diagrama de la arquitectura del Sistema

De esta forma, se observa el diagrama de la arquitectura mencionada anteriormente en la figura 4.16, como el sistema de este proyecto, sigue la arquitectura modelo - vista - controlador, y utiliza el dataframe como una pseudo base de datos, y la base de mongo como una base con información más completa sobre cada elemento.

De la misma manera, al observar la figura 4.17 se puede observar el funcionamiento más detallado del modelo, en el cual le entra la información, trae la información del dataframe, y realiza los cálculos necesarios para devoler una lista de costo mínimo de dietas diarias, de las que el controlador debe elegir la relacionada con las características del usuario.

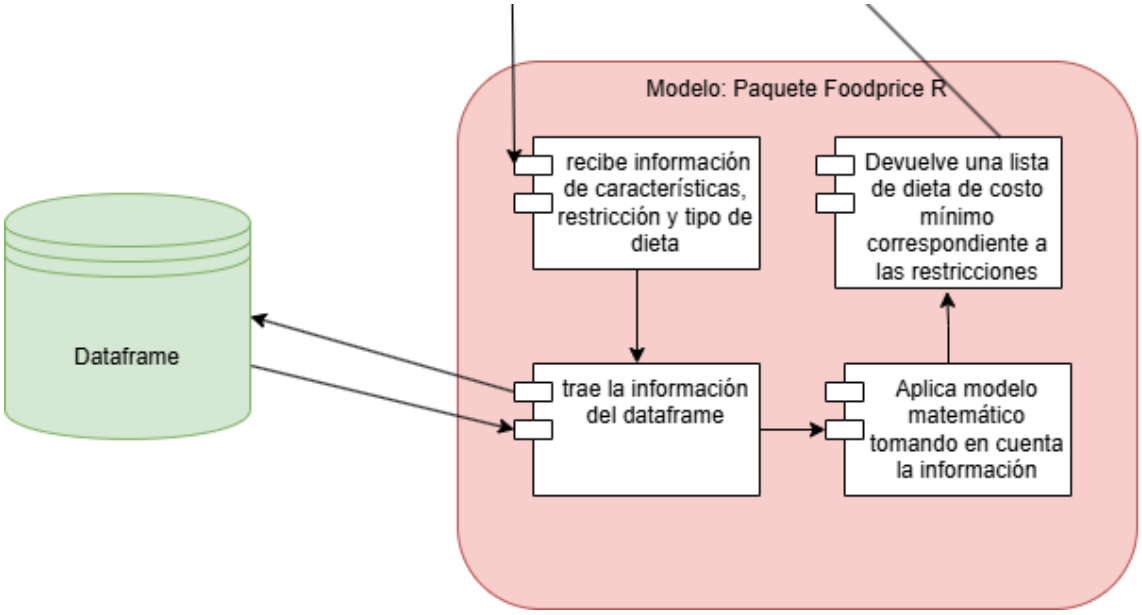

Figura 4.17: Diagrama de la aquitectura - modelo

Así mismo, se cuenta con la arquitectura detallada de la vista en la figura 4.18, aquí se observa como este elemento controla toda la interacción con el usuario, siendo aquel que muestra que pantalla se verá dependiendo del estado en el que se encuentre el sistema (se verá más a fondo en el diagrama de flujo en 4.21). Adedmás de esto, al interactuar con el usuario, es el que comnienza el proceso de seleccionar las diferentes características, restricciones, y tipo de dieta. Y al final recibe la dieta correspondiete para mostrarle al usuario.

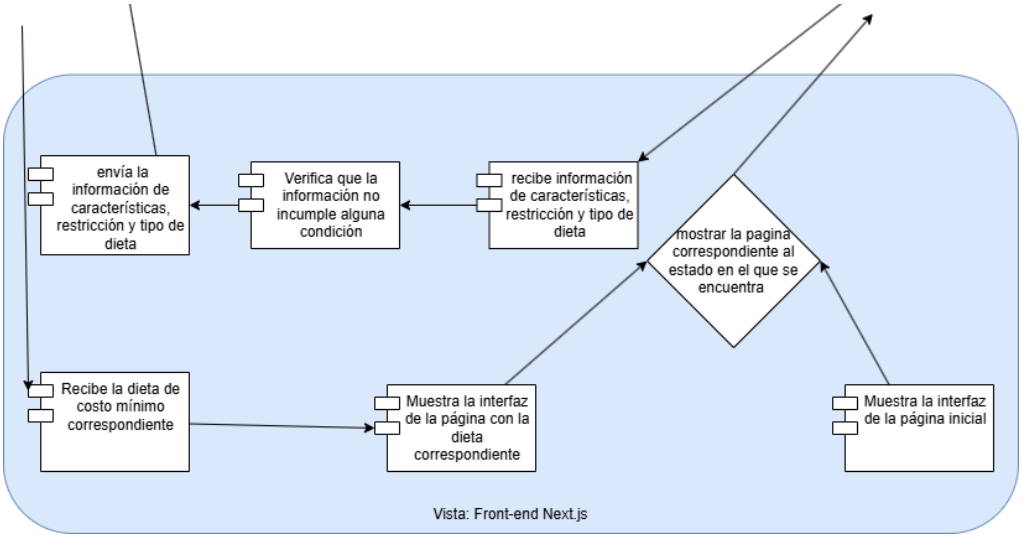

Figura 4.18: Diagrama de la aquitectura - vista

En el caso del controlador, en la figura 4.19 se observa como este recibe las entradas de la vista, las

transforma para funcionar de forma correcta con R, y las pasa al modelo; de aquí recibe una lista con dietas de costo mínimo, y termina de elegir la correcta, procede a relacionar sus alimentos con el listado completo de la base de Mongo, para entregarle todo eso a la vista de nuevo, y las pueda visualizar.

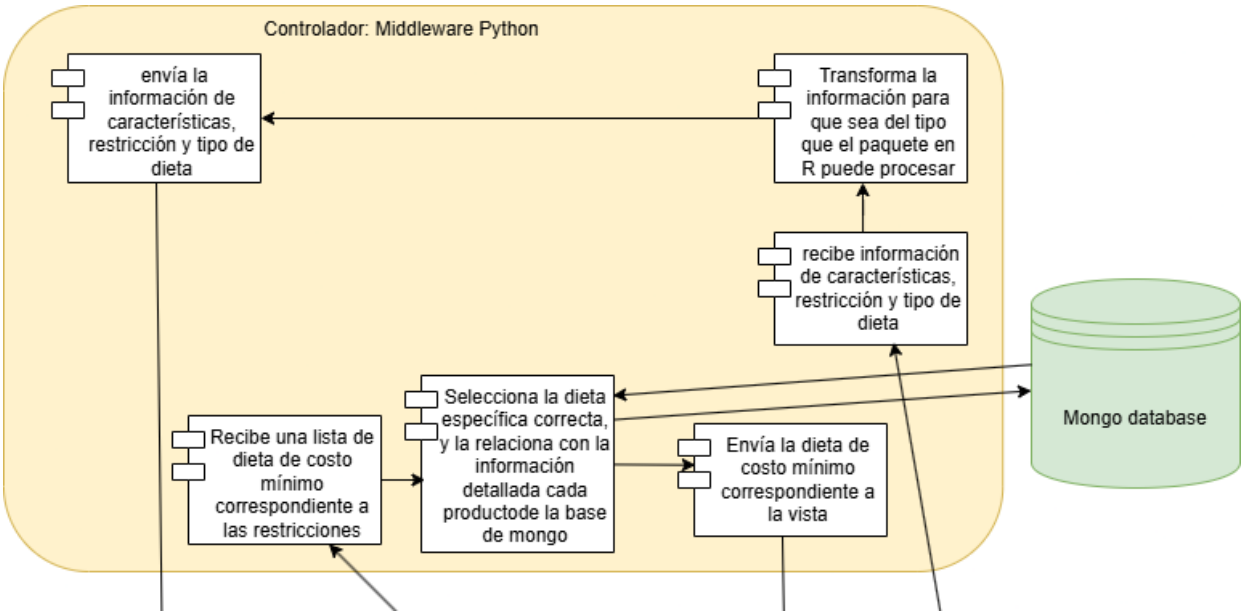

Figura 4.19: Diagrama de la aquitectura - controlador

Finalmente, como se ve en la figura 4.20 que muestra el diagrama detallada de la arquitectura del prototipo, se toma toda la información anterior.

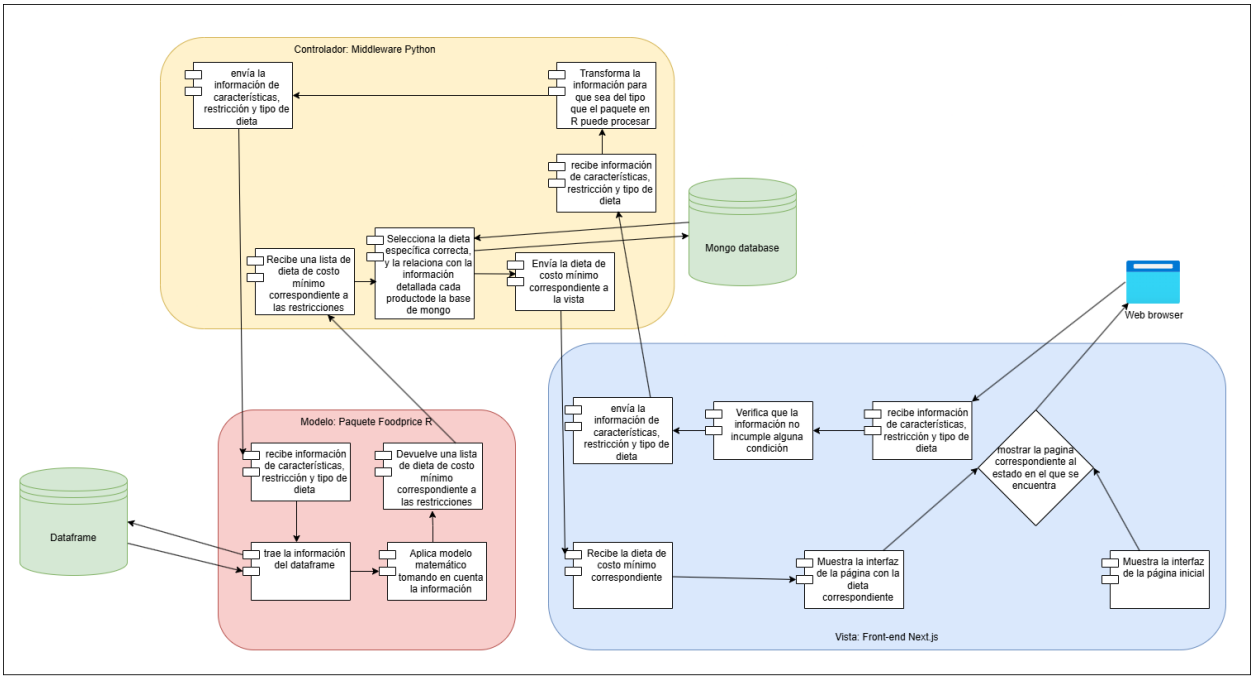

Figura 4.20: Diagrama de la arquitectura completa detallada

4.3.2. Diagrama de flujo

En el diagrama de flujo 4.21 se observa el flujo del proceso que puede tener el sistema en base a las acciones del usuario y la respuesta de este. Aquí, se contemplan todas los posibles caminos que puede tomar la aplicación, y se muestran los eventos que llevan a dichos caminos. De esta forma, se empieza con una pantalla inicial, de la cuál se pueden seleccionar 3 tipos de opciones a elegir en base a la opción con la que interactúe, y así se va navegando en la aplicación, como se muestra en el diagrama.

Vale la pena aclarar, que tal al llegar al estado de ver ya la dieta generada, puede decidir volver a la pantalla inicial para generar una nueva sin problema, por lo que el sistema no tiene un estado definitivo de final.

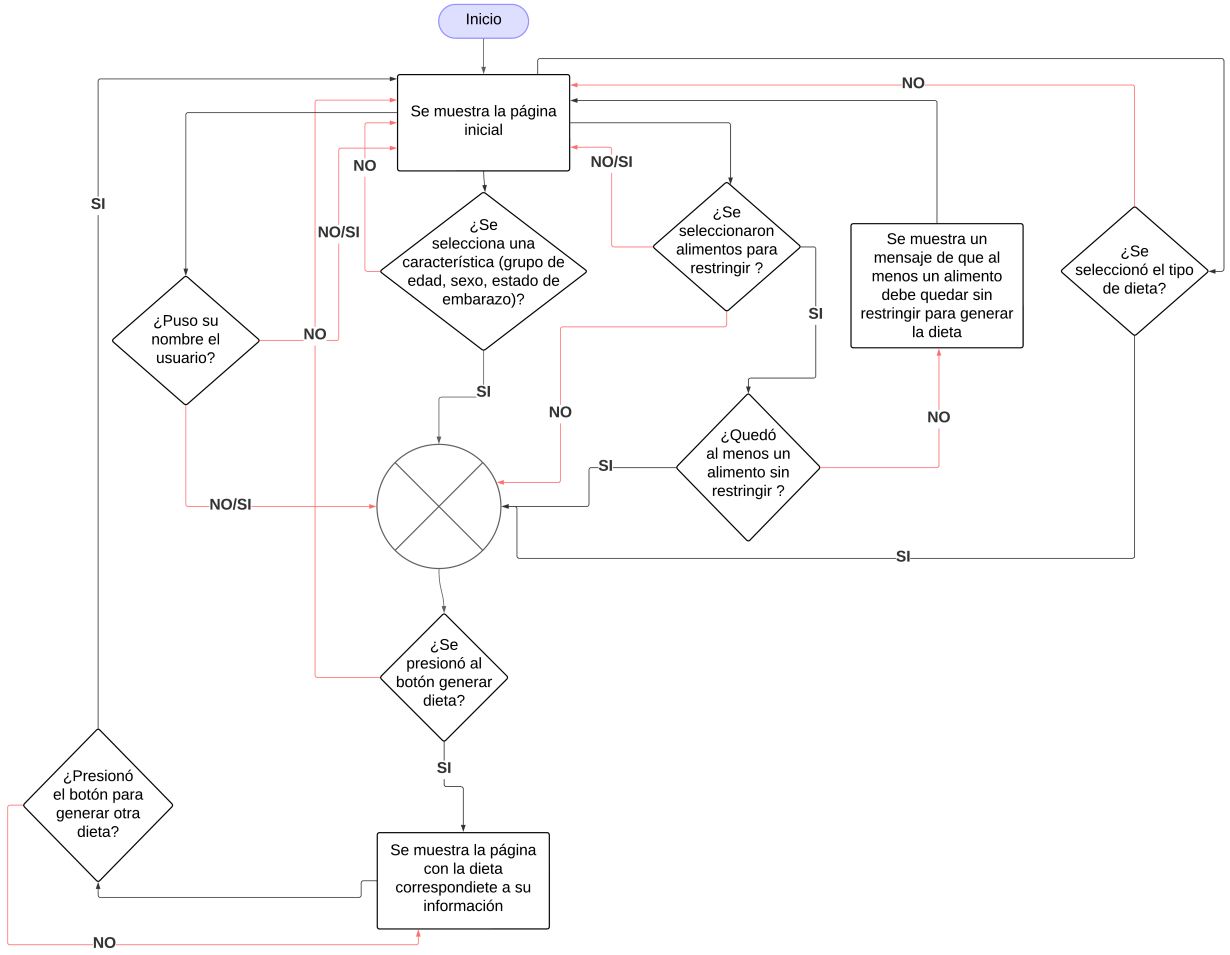

Figura 4.21: Diagrama de flujo del prototipo

4.3.3. Diseño del Modelo de Datos

En esta sección se aborda el modelo de datos que se manejó para el diseño del proyecto, tomando en cuenta que este proyecto consume tanto un paquete en R, como una API en Python, esto presentó retos a la hora del modelo de datos.

El más grande de estos retos fue el tener que usar dos modelos de datos de naturalezas distintas. Ya que el paquete en R interactúa mejor con un dataframe sencillo, del que pueda extraer los datos directamente. Y aunque el programa en Python podría trabajar con este dataframe, esto generaría dificultades debido a queries en base al id que se deben realizar en python, y pueden ser mucho más eficientes al realizarlas sobre una base de datos que permita el uso de ID's. Por este motivo, se decidió entonces tener como parte del sistema tanto un dataframe sencillo para ser consumido por R, como una base de datos más completa en MongoDB para la interacción con Python.

Otro de los grandes retos de este proyecto fue tener que adaptar los datos que se scrapearan del supermercado, para que encajaran con el modelo de datos del paquete en R. Para esto se tuvo que hacer un exhaustivo análisis de como el paquete Foodprice recibe sus datos, para poder adaptarlos.

Sin embargo, los modelos de datos relacionados con el uso directo del paquete en R, terminaron siendo más largas listas que no tenian un orden claro, ya que desde el paquete no se utilizó una estructura clara para los datos, habiendo incluso en ocasiones incongruencias entre los tipos de datos de un mismo campo. Además de que eran archivos de tipo .RDATA, por lo que no se podía interactuar con ellos como con una base de datos.

Debido a eso, se presenta aquí en la tabla 4.4 el modelo de datos que se usa en la base de datos de MongoDB que se desarrolló en este proyecto. Este modelo de datos se asegura de mantener un orden más claro, y sin cambios inesperados. En esta base de datos se guardaron los datos que se consiguieron con el proceso de web scraping, y estos solo se utilizan para interactuar con el sistema a través de python, relacionando cada alimento con su respectivo alimento en el dataframe que consume el proyecto en R.

Cuadro 4.4: Modelo de datos de alimentos

Campo	**Descripción**
_id	Identificador único del elemento dentro de MongoDB.
city	Ciudad de donde proviene el producto.
store	Tienda de donde proviene el producto.
name	Nombre del producto.
url	Enlace hacia el producto.
price	Precio del producto.
unit_price	Precio por cada 100 gramos de producto.
discount	Descuento con el que cuenta el producto.
image	Enlace hacia la primera imagen del producto.
timestamp	Hora a la que fue "scrapeado.[el] producto.
TCAC	Identificador del producto dentro de la base de datos de SIPSA (Sistema de Información de Precios y Abastecimiento).
subgroup	Subgrupo de alimentos al que pertenece el producto.

4.3.4. Diseño del prototipo

Para el diseño del prototipo, se pensó en un diseño simple e intuitivo, debido al nivel educativo de los usuarios, se decidió que no fuera un sistema complejo de usar, y que tenga textos que ayuden a guiar al usuario a través de la aplicación; así como también se tomó en cuenta que las letras sean legibles y los colores se diferencien fácilmente para así facilitar su uso.

Tomando en cuenta todos los aspectos anteriores, se creó el prototipo directamente en código, pero como un esqueleto sencillo sin funcionalidad real, solamente para tener claros algunos conceptos básicos como el estilo que tendría la aplicación, los colores, se definió blanco y negro, y los campos principales.

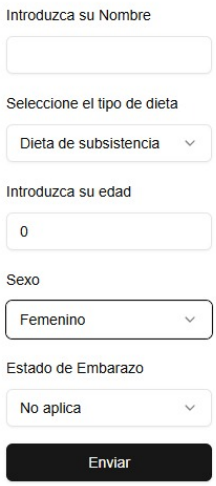

Figura 4.22: Prototipo de la aplicación

No se hicieron tantos prototipos de todas las pantallas ya que al hacer un esqueleto sencillo en código, se propiciaba el querer avanzar a la siguiente fase para darle funcionalidad final a todo y tomarse su tiempo con cada componente. Por lo que la fase de diseño fue más bien corta, pero sirvió de base para en la implementación tener claro el estilo que se seguiría en la aplicación en general

Figura 4.23: Prototipo 2 de la aplicación

Sin embargo al ver el resultado final de la aplicación se puede apreciar el avance que hubo desde los prototipos, sobre todo a nivel de facilidad de acceso a la información y explicación de cada elemento, teniendo en cuenta que los usuarios no cuentan con conocimiento previo de la aplicación.

Capítulo 5

Implementación

5.1. Dispositivos de desarrollo y despliegue

Con el fin de desarrollar la herramienta se utilizó un computador de mesa con las siguientes características:

- Computador
  - **Tarjeta Gráfica:** GeForce RTX 3060
  - **Procesador:** AMD Ryzen 5 3400G
  - **RAM:** 16 GB
  - **Almacenamiento:** 1 TB

Se eligió trabajar con un computador que tuviern una buena tarjeta gráfica y procesador decente, para hacer del proceso de desarrollo más fluido y sin complicaciones.

Para hacer pruebas también se utilizó un celular:

- Celular
  - **Modelo:** Xiaomi Redmi Note 10
  - **RAM: 8GB**
  - **Procesador:** Snapdragon 732G
  - **GPU:** Adreno 618
  - **Cámara Frontal:** 16 MP, f/2.5

5.1.1. Proceso de conectar Python con R

El proyecto Foodprice es un proyecto de la Pontificia Universidad Javeriana Cali que busca generar un sistema que permita a las personas en condición de pobreza generar una dieta de costo mínimo que se adapte a sus necesidades nutricionales, así como a sus características físicas. Sin embargo dicho proyecto presenta una complicación, al estar desarrollado en el lenguaje R, es muy poco accesible para la población en general, y menos aún para las poblaciones de escazos recursos, esto debido a que estadisticamente estas poblaciones son más propensas a tener bajo nivel de escolaridad [referencia], por lo que un software que requiera conocimiento técnico alto para usarse, será complicado

que logre tener un impacto.

Por este motivo, se crea este proyecto. Sin embargo, en su desarrollo también se presentan diversos obstáculos. Esto debido a que se quiere poder consumir el paquete en R de Foodprice a través de un lenguaje más moderno y con un gran alcance como lo es Python. Pero, lograr comunicar de forma fluida y eficiente aplicaciones construidas en lenguajes diferentes no resulta ser una tarea fácil, por lo que se optó por convertir el paquete de Foodprice en una API, para así interactuar de estar forma con una API en python, la cual le pase una entrada, que el paquete en R utilice para calcular la dieta de costo mínimo adecuada a sus necesidades. Para poder lograr esto, se tuvo que modificar también el paquete en R, para recibir parámetros externos y calcular las dietas con normalidad. Además de esto, se utilizó el paquete Plumber para lograr convertir Foodprice en una API, y se utilizó FastAPI en Python para crear la API de python que se encargará de llamar al proyecto Foodprice y enviarle los parámetros para que funcione adecuadamente.

5.1.2. Retos de implementación backend (Web Scraping)

Ya que el paquete Foodprice realiza cálculos de dietas basado en precios mayoristas aportados por el Departamento Administrativo Nacional de Estadística, dichos datos no reflejan de manera acertada los precios a granel de una dieta que un ciudadano común podría encontrar en un supermercado. Con la finalidad de contrarrestar esta falencia, se decidió emprender un proyecto de web scraping en el supermercado Almacenes Éxito; donde se buscó obtener la información de múltiples productos que se encontraran listados en el sistema de Guías Alimentarias Basadas en Alimentos (GABAS) del Instituto Colombiano de Bienestar Familiar.

Esto debido a que el GABAS provee los datos nutricionales de los alimentos, con lo cual al obtener los productos de un supermercado común y cruzar la información con esta base de datos, se obtendría un mejor estimado en cuanto a la nutrición que se puede obtener por cada peso que se invierte en la alimentación de distintos grupos poblacionales. Sin embargo, no siempre los nombres de los productos en GABAS o SIPSA se corresponden a los nombres coloquiales o comunes por los cuales se consiguen en los supermercados. Tomemos como ejemplo el producto con identificador C079 en la base de datos SIPSA, este producto tiene el nombre de “Patilla”, sin embargo el uso de este nombre no es tan común como el de “Sandía”, motivo por el cual si se realizara un proceso de web scraping sin cuidado, se podría llegar a casos en los que no se encuentre información de ciertos productos cuando simplemente buscando con otros nombres estos aparecerían.

Una vez identificada esta problemática, es claro que se tiene que realizar un proceso de verificación de los datos obtenidos por el sistema de web scraping con la finalidad de obtener y pulir la mayor cantidad de datos posible; esta verificación manual puede resultar exhaustiva, pero vale la pejna cuando se piensa en la meta de generar dietas con datos precisos y realistas, que suplan las necesidades de poblaciones económicamente vulnerables por el menor costo posible.

5.1.3. Retos implementación frontend

Ya que el proyecto tiene como objetivo ofrecer un sistema accesible para el cálculo de dietas, se tiene que asegurar que el sistema se adapte bien a todo tipo de pantallas, desde las más pequeñas hasta las más grandes, lo cual supone un reto cuando se quiere contar a la vez con un sistema agradable a la vista.

Otro factor clave para este proyecto era la población objetivo, los cuales al encontrarse en situac ión de pobreza, tienen mayor probabilidad de no tener acceso a un alto nivel eduativo, tecnología de alta gama, o un internet de alta velocidad. Debido a esto se priorizó que el diseño final de la aplicación fuera simple, sin tantos elementos cargados que puedan confundir, con una navegación directa, y con textos que acompañen y expliquen las funcionalidades del sistema.

Con todo esto, la aplicación final se observa a continuación:

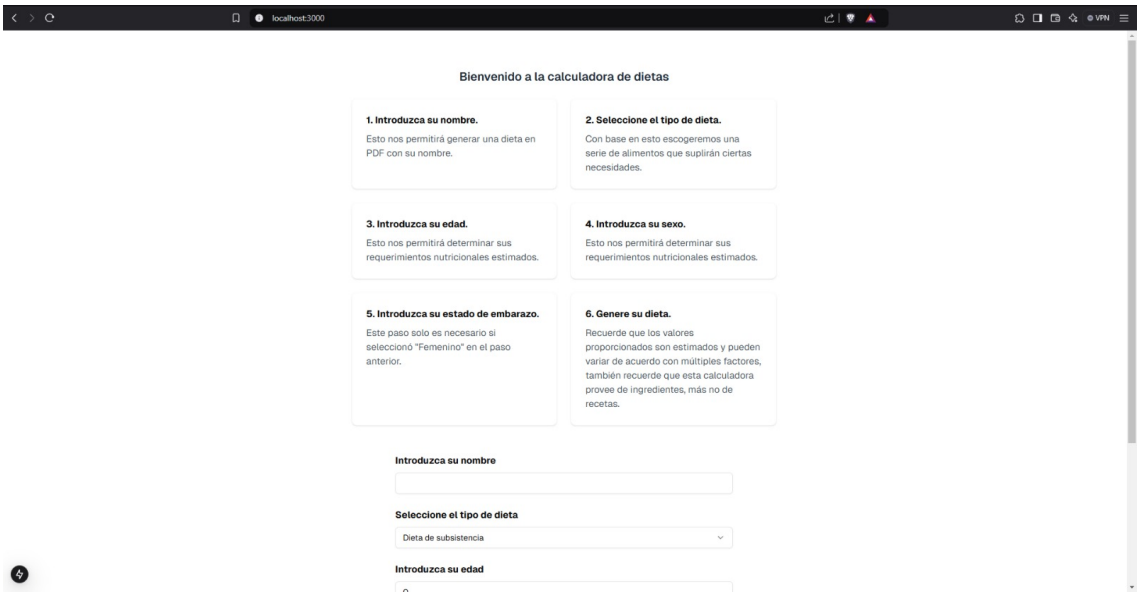

Figura 5.1: implementación del sistema 1

Como se puede observar en la figura 5.1, es la vista inicial que se tiene del sistema al utilizarlo en PC, aquí se puede observar como un factor importante es dar instrucciones claras, para que el usuario entienda que hacer, y sepa también que esperar

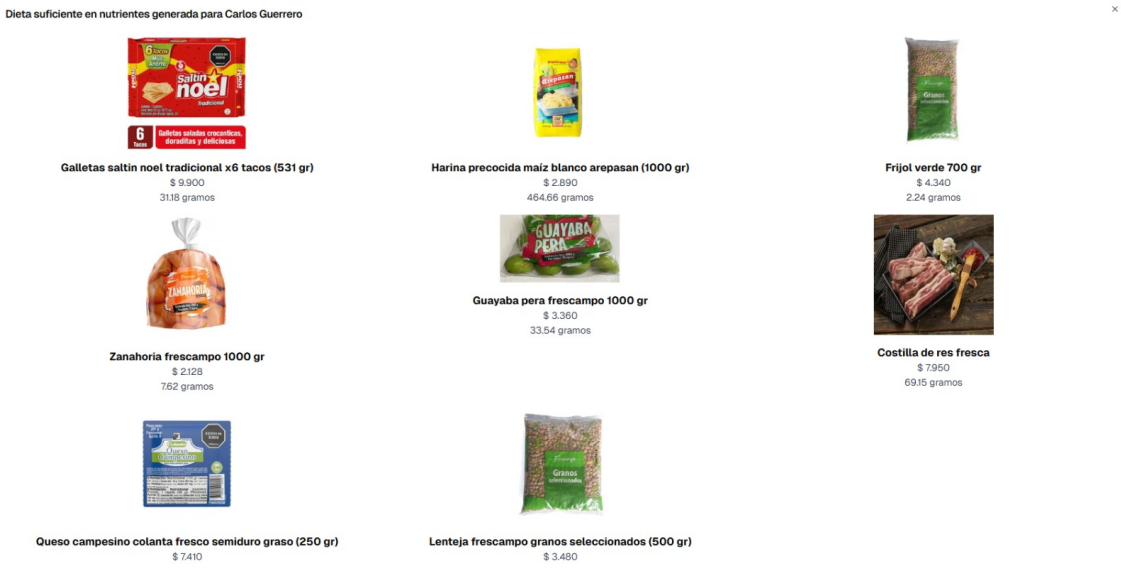

Figura 5.2: implementación del sistema 2

De la misma forma se observa en la figura 5.2, la vista de una dieta ya generada, adaptada a las necesidades del usuario, en la cual destaca el uso de imágenes para mostrar los alimentos, con lo cual se puede realizar una asociación más directa a través de la vista que solo al leer una lista de productos. Se observa también como se ve directamente toda la informaciń relevante respecto a cada producto, y además si se presiona cualquiera de los productos, dirige al link para comprar dicho producto.

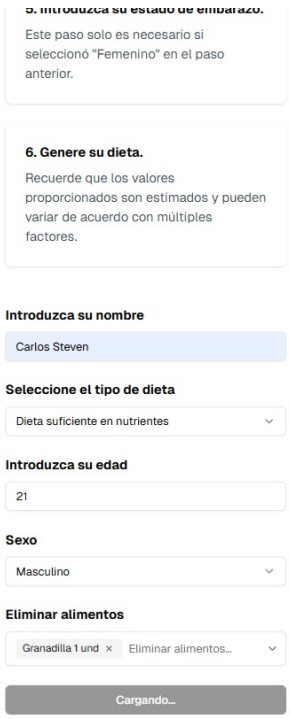

Figura 5.3: implementación del sistema 3

En comparación, se tiene también en la figura 5.3 la aplicación al verla desde un teléfono movil, con lo cual a pesar de que cambian las dimensiones, se sigue manteniendo la estética del proyecto, además de que se conservan los elementos más importantes, la información para generar una buena experiencia de usuario.

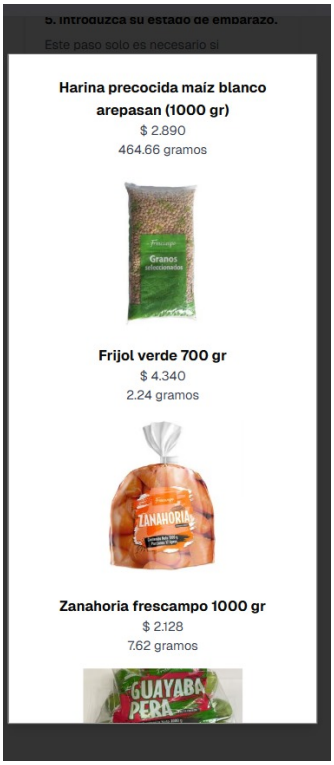

Figura 5.4: implementación del sistema 4

Figura 5.5: implementación del sistema 5

Capítulo 6

Pruebas

Para este proyecto, se decidió centrar las pruebas en dos tipos en específico, pruebas de carga y pruebas de aceptación. Esto debido a una limitante de tiempo, se eligieron entonces estas cargas ya que es vital garantizar que el sistema pudiera manejar m'ultiples solicitudes simult'aneamente sin comprometer su desempeño. Al igual que es de suma prioridad asegurar que el sistema cumpliera con los requisitos funcionales y no funcionales definidos inicialmente, y que estos satisfacieran las expectativas del usuario final.

6.1. Pruebas de carga

Haciendo uso del framework de pruebas Locust, que se enfoca en las pruebas de carga sobre aplicaciones web, se probó simular quince usuarios simultáneos enviando múltiples solicitudes de dietas suficientes en nutrientes (CoNA) con diferentes edades, esto para probar el desempeño del sistema en un escenario realista.[en] el cual se cuenta con distintos usuarios de distintas edades solicitando dietas a la vez.

Figura 6.1: Prueba de carga

En los resultados de la prueba de rendimiento, ejecutada durante un período de cinco minutos con

una carga constante de usuarios, se observó que los tiempos de respuesta del sistema experimentaron latencias significativamente altas en la mayor parte del análisis. Específicamente, cuando la tasa de solicitudes superó 1 petición por segundo, los tiempos de respuesta oscilaron entre 10.000 ms y 30.000 ms.

Este comportamiento podría atribuirse a la arquitectura del sistema, en la que los cálculos no se realizan en el mismo servidor que maneja las solicitudes de la aplicación. En particular, el procesamiento se delega al servidor Plumber, que opera con el paquete Foodprice, mientras que la aplicación principal funciona sobre FastAPI. Esta comunicación interservidor introduce una latencia adicional, la cual parece ser un factor determinante en los picos elevados de tiempo de respuesta observados.

Para mejorar la experiencia del usuario y cumplir con los requisitos de rendimiento, se recomienda optimizar la comunicación entre FastAPI y Plumber, reducir la latencia en la transmisión de datos y evaluar la posibilidad de distribuir la carga de procesamiento de manera más eficiente.

6.2. Pruebas de aceptación

Se realizaron pruebas con usuarios, con la colaboración de 7 conocidos y 8 personas desconocidas a través de diferentes días en los que se tuvo la oportunidad de que probaran la aplicación web de este proyecto; se les explicó de que va el proyecto, y lo básico de su funcionamiento, pero se les incitó a que intentaran usar el sistema solo guiándose con las instrucciones de la aplicación, y después de utilizarla corriendo en mi computador personal, se les pidió que llenaran una encuesta para conocer su opinión frente al sistema.

Estas pruebas se hicieron a través de una encuesta de google, en la que todos sus datos eran anónimos, para que tuvieran libertad de dar su opinioón sincera.
A continuación se plasman los resultados de dicha prueba:

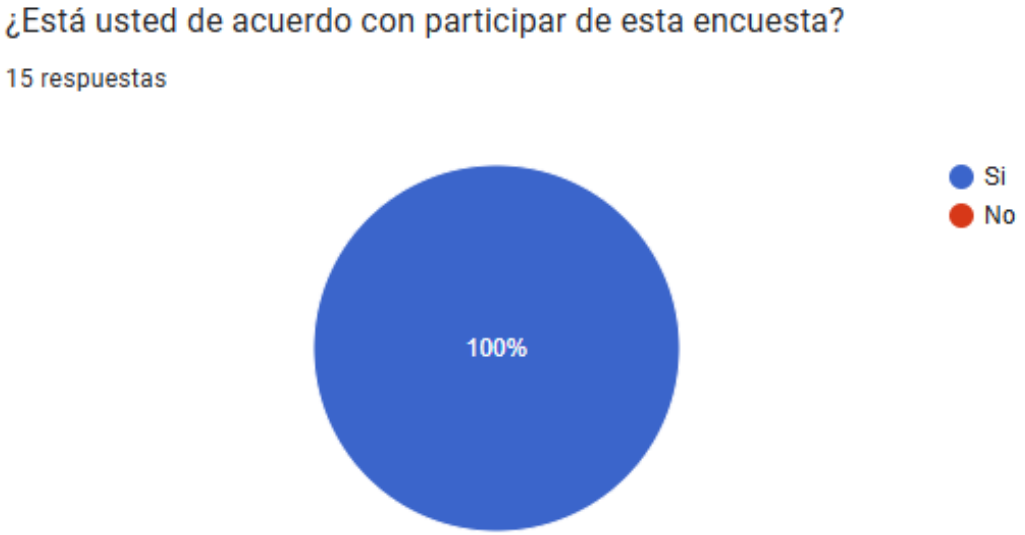

Figura 6.2: Prueba de aceptación 1

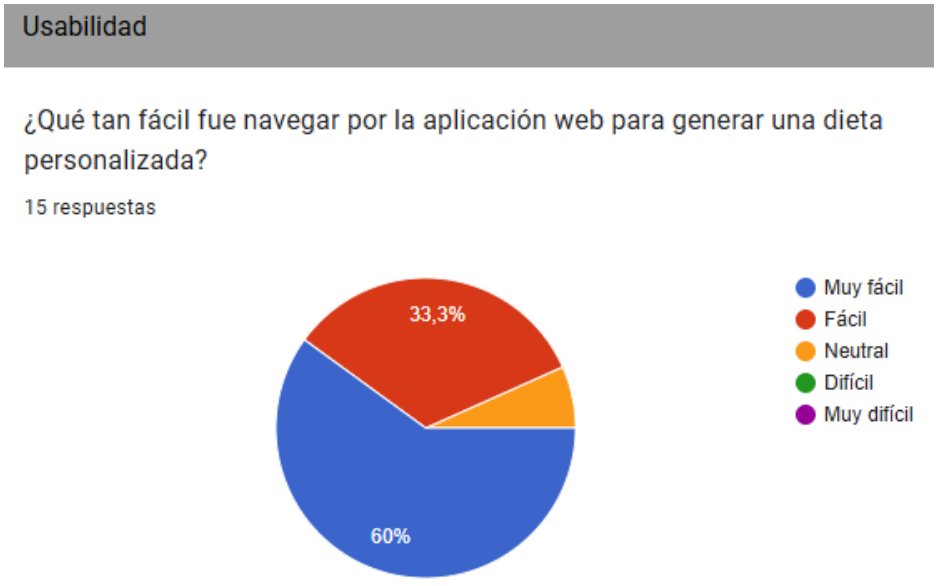

Figura 6.3: Prueba de aceptación 2

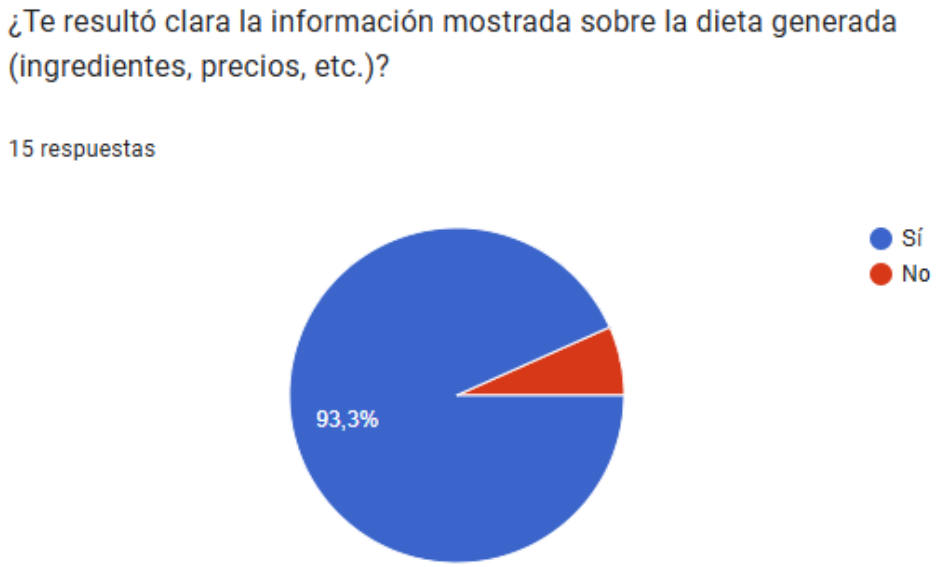

Figura 6.4: Prueba de aceptación 3

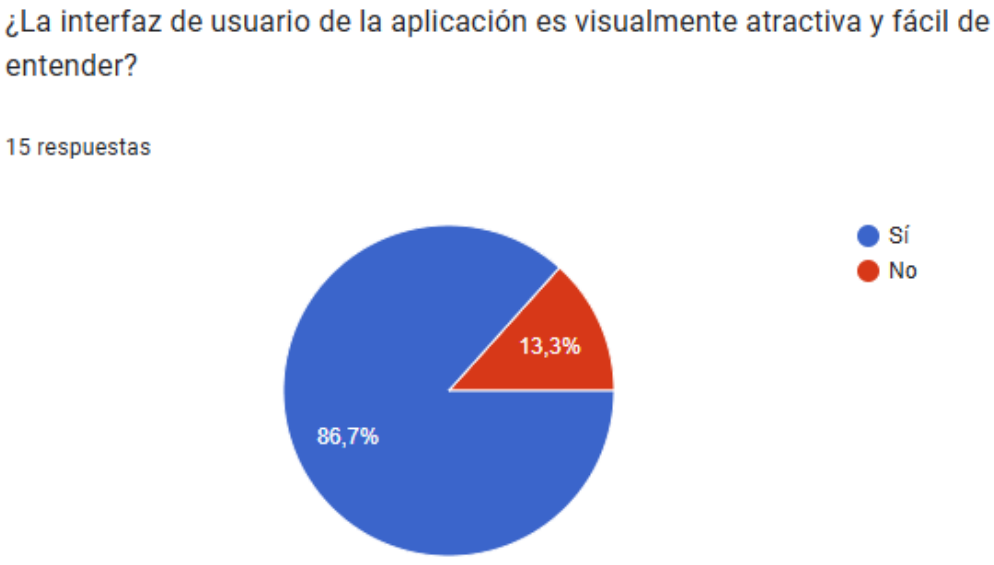

Figura 6.5: Prueba de aceptación 4

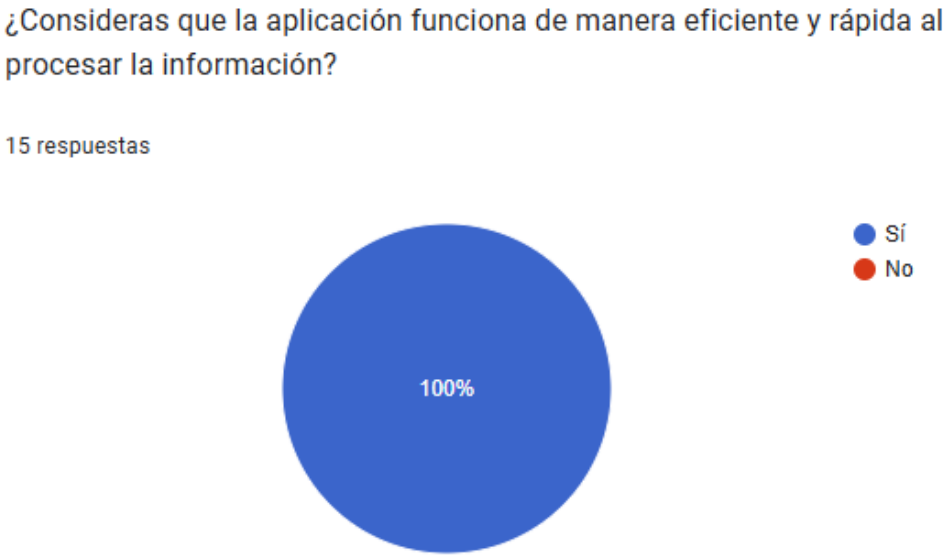

Figura 6.6: Prueba de aceptación 5

Figura 6.7: Prueba de aceptación 6

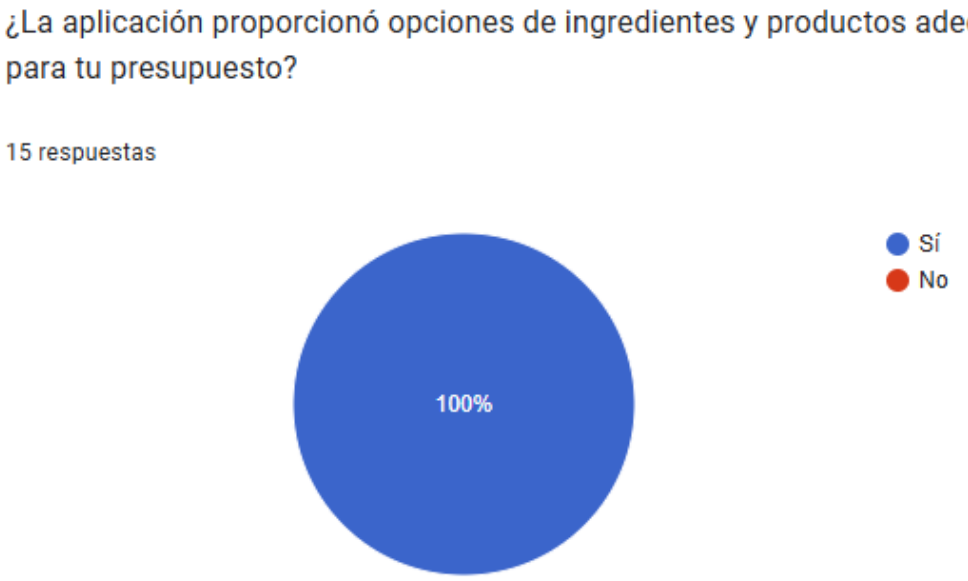

Figura 6.8: Prueba de aceptación 7

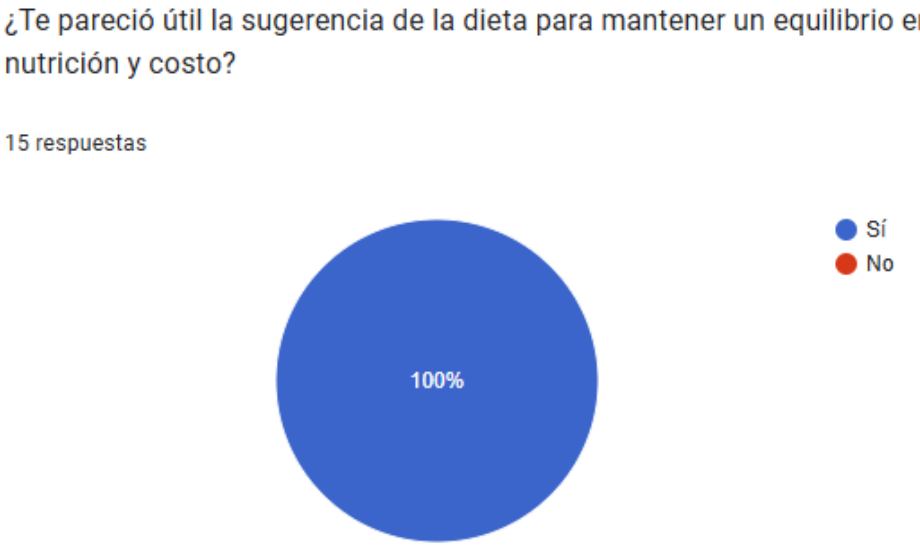

Figura 6.9: Prueba de aceptación 8

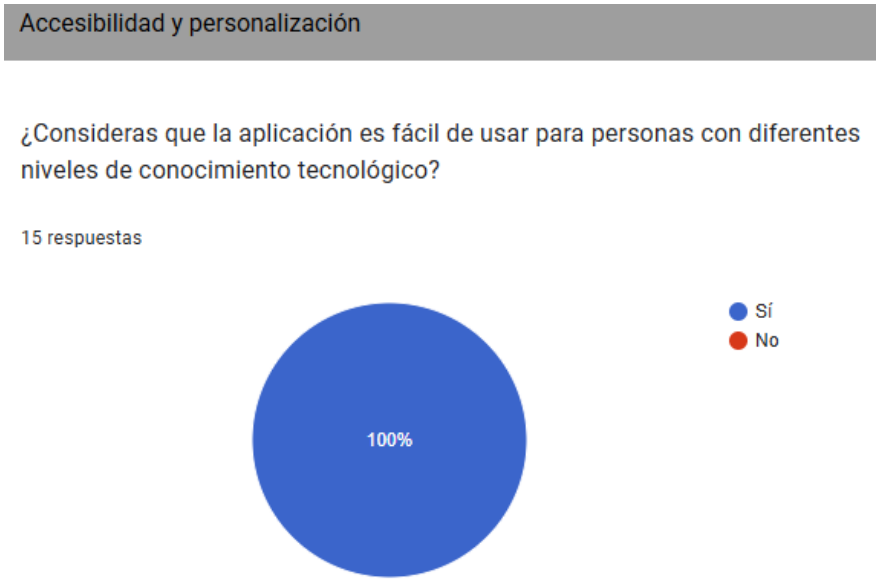

Figura 6.10: Prueba de aceptación 9

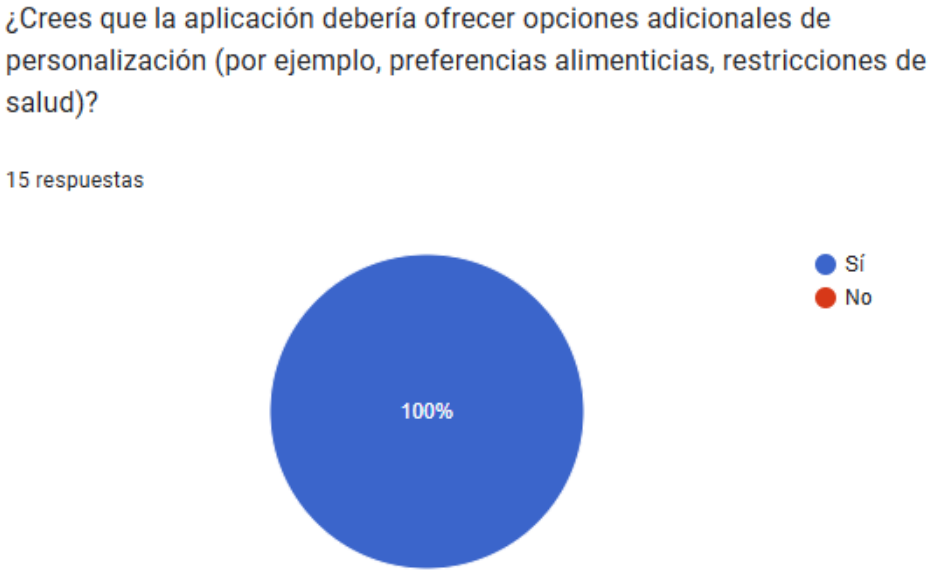

Figura 6.11: Prueba de aceptación 10

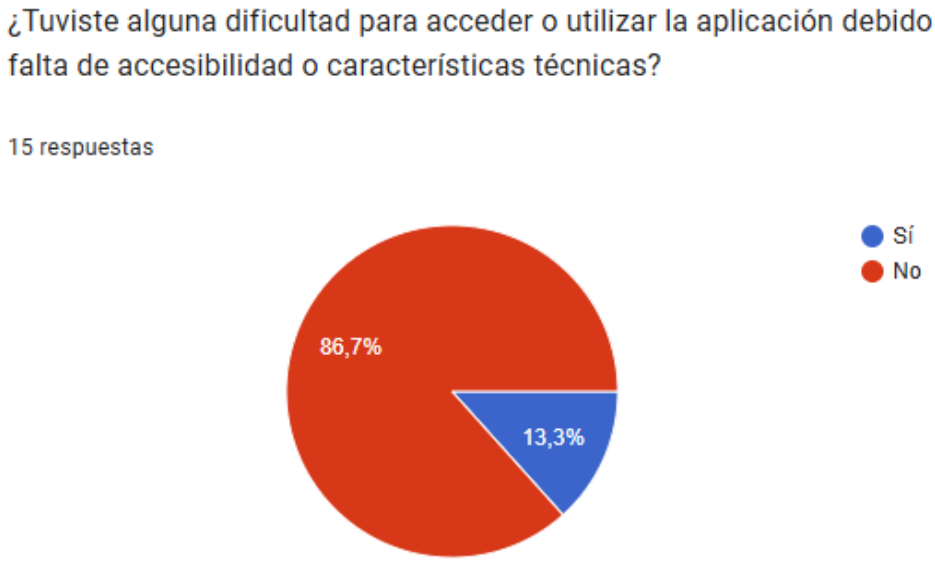

Figura 6.12: Prueba de aceptación 11

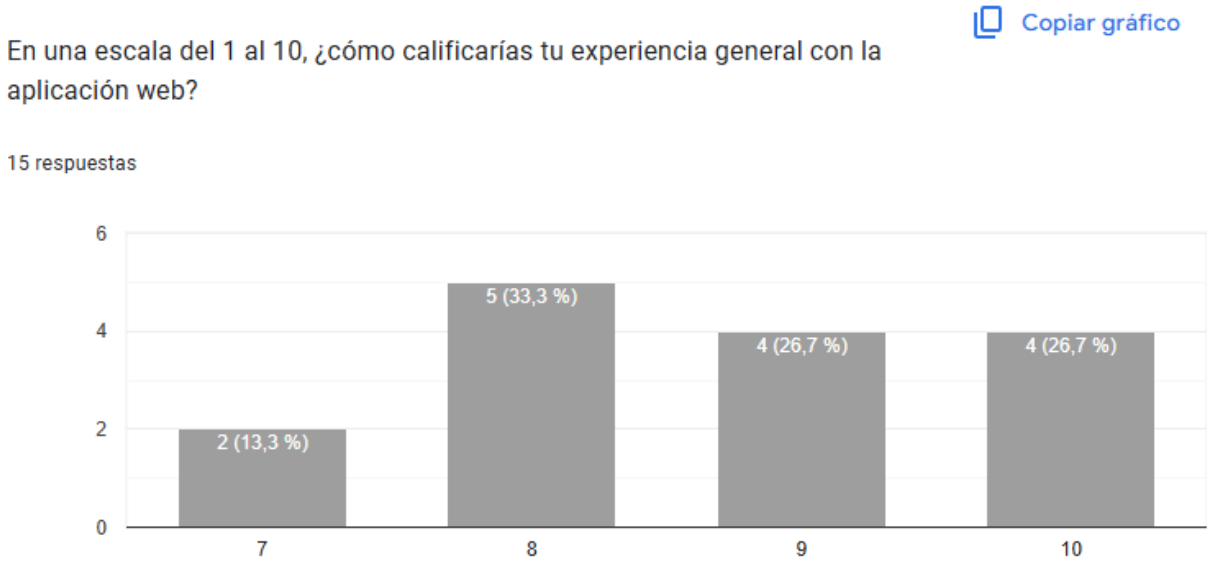

Figura 6.13: Prueba de aceptación 12

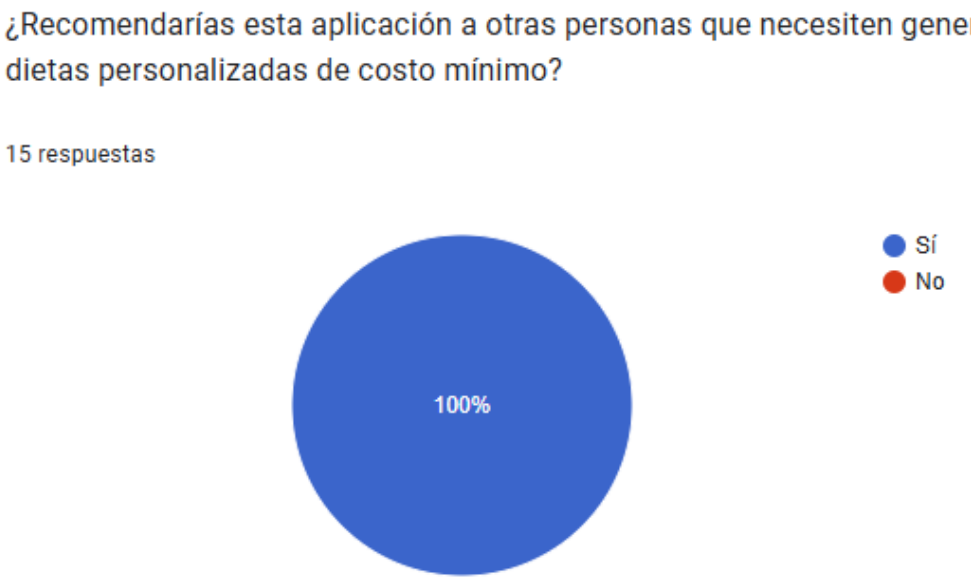

Figura 6.14: Prueba de aceptación 13

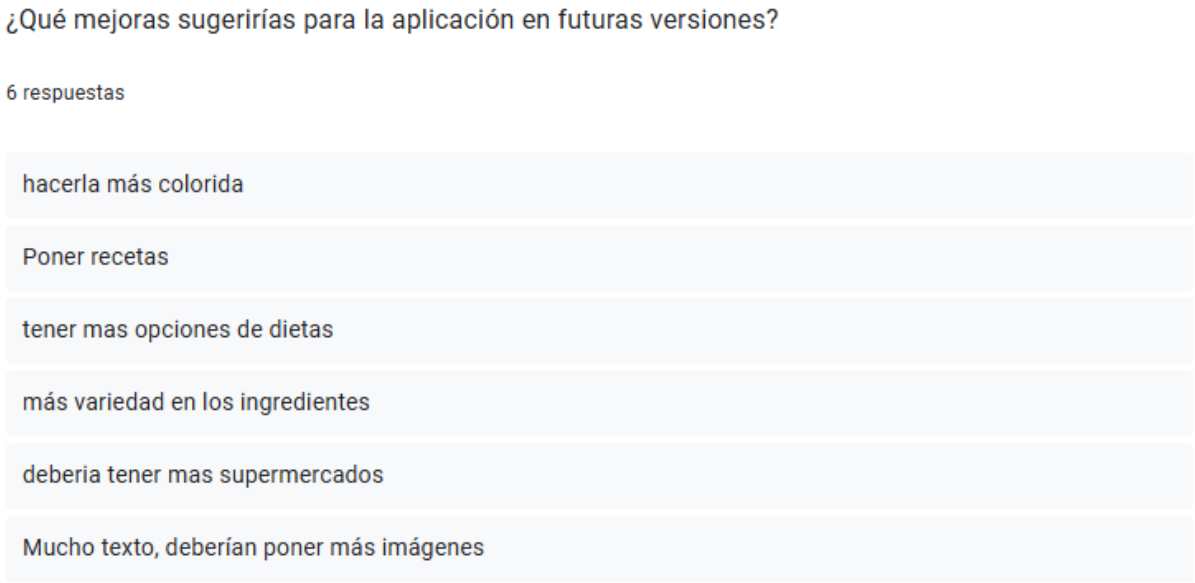

Figura 6.15: Prueba de aceptación 14

Con la información recolectada a través de estas pruebas, se puede observar que el sistema tiene un nivel de aceptación bastante alto, teniendo altos porcentajes de respuestas positivas frente a preguntas como la facilidad de navegación, la calificacón que le dan, entre otras.

Y las sugerencias son relacionadas no se centran en general en señalar algo negativo que tenga

el sistema, sino formas en las que les gustaría a esos usuarios que el sistema incluyera más cosas.

A futuro, sería ideal poder realizar pruebas de este estilo con una población que represente realmente la población que se quiere impactar con este proyecto; para así medir realmente cual es el impacto en la vida de las personas.

Capítulo 7

Despliegue

Para la fase dedespliegue, se explicará como es el proceso para desplegar el sistema.

7.1. Instalación de Foodprice

Se debe realizar la clonación del paquete desde GitHub.

git clone https://github.com/imcarlosguerrero/FoodpriceR

Desde RStudio abrimos el archivo **FoodpriceR.Rproj** que pondrá a punto el entorno de trabajo sobre el cual se ejecutará el paquete.

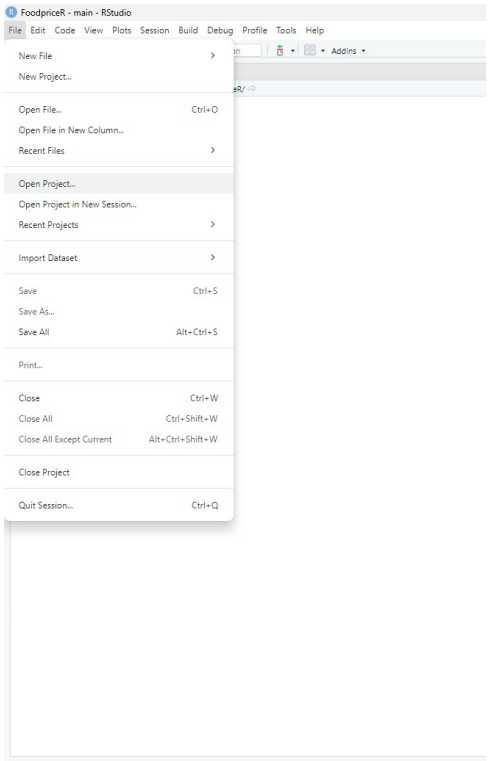

Figura 7.1: Despliegue del sistema

Figura 7.2: Despliegue del sistema 2

En la consola de RStudio ejecutamos el siguiente comando.

devtools::install("../FoodpriceR")

Este instalará el paquete Foodprice de manera local mientras nos encontramos dentro de la ruta de trabajo.

El paquete Foodprice requiere que en el entorno de trabajo se encuentren cargados los siguientes dataframes.

- Mapeo_Sipsa_TCAC
- Mapeo_Sipsa_TCAC_GABAS_Grupos
- Mapeo_Sipsa_TCAC_Carga_2
- Primer_Criterio_Lista_Alimentos
- intercambio_gramos
- TCAC

Estos dataframes los podremos encontrar en la carpeta data que se encuentra dentro del proyecto y la cual podemos acceder desde el navegador de archivos de RStudio que se encuentra en la zona derecha del entorno de desarrollo.

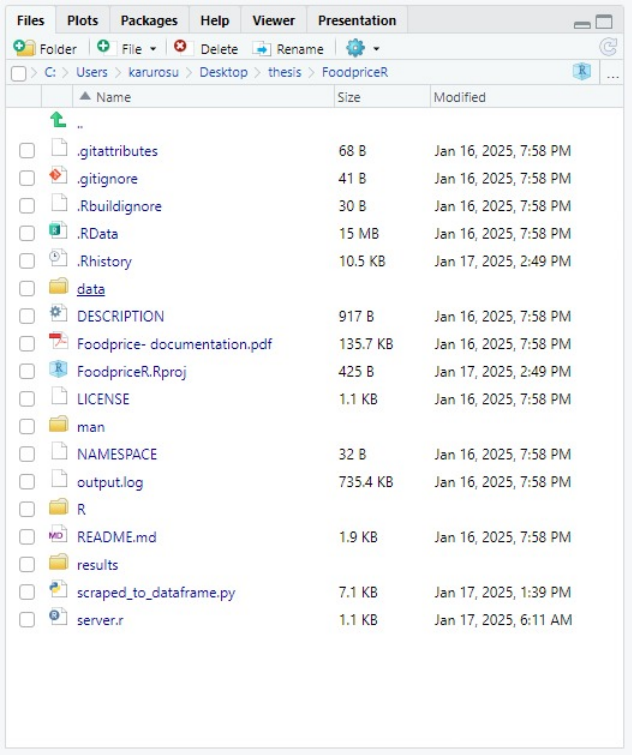

Figura 7.3: Despliegue del sistema 3

Una vez hayamos dado click sobre el dataframe que deseamos cargar, se abrirá una ventana emergente preguntando si queremos cargar dicho dataframe al entorno global, presionamos **"sí"**.

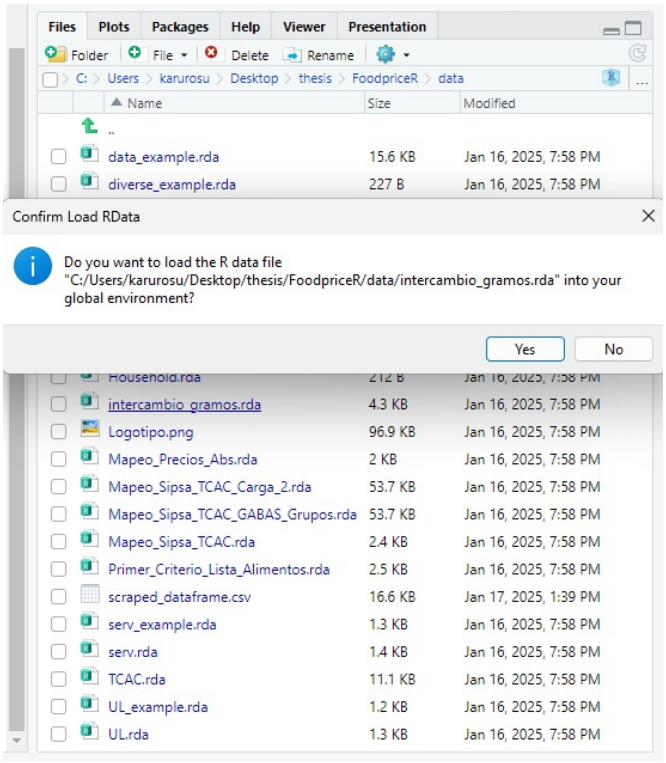

Figura 7.4: Despliegue del sistema 4

y con esto ya tendremos cargado uno de los dataframe que el paquete Foodprice requiere para su funcionamiento, repitiendo este proceso por la cantidad de dataframes restantes, completaremos todos los requerimientos para poder comenzar a utilizar las funciones del paquete.

Capítulo 8

Mantenimiento

Una vez observada la problemática de la alta latencia cuando se realizan múltiples solicitudes, nace la necesidad de realizar un desarrollo adicional que solucione esto, con lo que se propone como trabajo de mantenimiento la reivisión a fondo de la forma en la que el paquete Foodprice envía las dietas con la finalidad de reducir la complejidad de las operaciones y lograr mejores tiempos de respuesta.

Capítulo 9

Conclusiones

Con los resultados obtenido en este proyecto, se evidencia que si es posible el desarrollo de un sistema accesible para la generación de dietas saludables de costo mínimo con consideraciones de economía y nutrición personalizadas.

Sin embargo, a pesar de haber cumplido el objetivo principal de este trabajo de grado, se destaca como el contar con una base méjor estructurada hubiera facilitado el desarrollo, debido a que muchas de las complicaciones que se tuvo durante la elaboración fueron por malas prácticas de programación y de estructuración que se encontraban en el proyecto base; a pesar de esto, se logró crear una plataforma que cuenta con una interfaz con buena accesibilidad, y que permite visualizar de manera gráfica y con precios realistas una dieta que cumpla con las diferentes necesidades de las personas.

El sistema permite la creación desde dietas de subsistencia hasta dietas enteramente saludables y con variedad de alimentos, donde inclusive se permite la eliminación de productos que puedan no ser del agrado del usuario o que de plano no puedan consumir, ya sea por una afección médica, alergias o por simple disgusto hacia un alimento en particular, con lo que la construcción de una dieta saludable y personalizada es algo completamente viable con el sistema desarrollado.

Sin embargo, aun quedan dilemas por resolver para que el sistema pueda funcionar a toda su capacidad, no se logró que la base de datos se actualice de forma constante, debido al gasto computacional que esto requiere por el proceso de web scraping, por lo que el proyecto todavía tiene mucho potencial de crecer y mejorar.

Capítulo 10

Trabajo Futuro

Si bien se logró el objetivo de desarrollar un sistema accesible, existen características de este que pueden ser mejoradas, se plantea como trabajo futuro lo siguiente.

- La optimización del paquete Foodprice para entornos de conexión entre servicios.
- La optimización del sistema de web scraping para poder realizar obtención de datos de manera periódica, con lo que se podría asegurar precios siempre actualizados en el sistema.
- Agregar opciones de personalización como peso y altura para que la personalización de la dieta se incluso mayor a la actual que se encuentra basada en promedios obtenidos del Departamento Administrativo Nacional de Estadística.
- Integrar una mayor variedad de alimentos entre las opciones, lo que permitiría a los usuarios contar con mayores opciones cuando hayan repetido una receta basada en ciertos ingredientes.
- Siguiendo el punto anterior y contando con más opciones de alimentos se propondría el desarrollo de una calculadora de minuta semanal donde cada día se cuente con distintas combinaciones respecto a los demás días de la semana, permitiendo al usuario variar su alimentación.

Bibliografía

[1] G. E. S. Sarmiento, "Desnutrición en colombia - desde lo social, lo económico, y lo político,"

[2] Objetivos de Desarrollo Sostenible, "Hambre cero - la agenda 2030 en colombia - objetivos de desarrollo sostenible." Disponible en https://ods.dnp.gov.co/es/objetivos/hambre-cero.

[3] R. M. Prenk, "Aumenta desnutrición infantil en colombia." Disponible en https://www.senado.gov.co/index.php/el-senado/noticias/5443.

[4] M. F. Gutiérrez, "La relación entre el factor económico y la mala alimentación." Disponible en https://www.javeriana.edu.co/pesquisa/el-factor-economico-es-el-culpable-de-una-mala-alimentacion/.

[5] R. Burgos Peláez, "Desnutrición y enfermedad," *Nutrición Hospitalaria*, 2013.

[6] Fundación Éxito, "Desnutrición y pobreza van de la mano. ¿Cómo combatirlas? Escuchemos al experto John Hoddinott." Disponible en https://blog.fundacionexito.org/noticias/desnutricion-y-escuchemos-al-experto-john-hoddinott.

[7] G. D. Restrepo, L. F. R. Betancur, and J. E. V. Vargas, "Muertes por desnutrición en américa del sur en los últimos veinte años," vol. 18, no. 34, pp. 95–107.

[8] Organización Mundial de la Salud, "Nutrición." Disponible en https://www.who.int/es/health-topics/nutrition.

[9] Organización Mundial de la Salud, "Dieta sana." Disponible en https://www.who.int/es/health-topics/healthy-diet.

[10] Dane, "Boletín pobreza monetaria en la niñez y adolescencia en colombia." Disponible en https://www.dane.gov.co/files/lineas-de-tiempo/pobreza-monetaria-ninez-adolescencia-en-colombia/.

[11] Amazon AWS, "What is a web application?." Disponible en https://aws.amazon.com/what-is/web-application/.

[12] M. Shaheel, "Diet recommendation system using machine learning," 12 2023.

[13] C. Lee, S. Kim, C. Lim, J. Kim, Y. Kim, and M. Jung, "Diet Planning with Machine Learning: Teacher-forced REINFORCE for Composition Compliance with Nutrition Enhancement," in *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, pp. 3150–3160, ACM.

[14] M. Nelson, K. Dick, and B. Holmes, "Food budget standards and dietary adequacy in low-income families," vol. 61, no. 4, pp. 569–577.

[15] S. Laoyan, "Qué es la metodología waterfall y cuándo utilizarla." Disponible en https://asana.com/es/resources/waterfall-project-management-methodology.

[16] "Functional vs non functional requirements," Apr 2020.

[17] PlaSA Colombia, "Acerca de plasa colombia." Disponible en https://plasacolombia.com/nosotros/.

[18] J. T. Baumgartner, "Congreso mundial de nutrición 2021." Disponible en https://www.diet-creator.com/es/noticias/congreso-mundial-de-nutricion-2021.

[19] Brave, "Web scraping meaning & definition." Disponible en https://brave.com/glossary/web-scraping/.

[20] K. Palacios, "Los 40 supermercados más grandes de colombia." Disponible en https://america-retail.com/paises/colombia/los-40-supermercados-mas-grandes-de-colombia/.

[21] Brave, "El "hard discount", más que un formato de retail, una filosofía." Disponible en https://panamericanlatam.com/el-hard-discount-mas-que-un-formato-de-retail-una-filosofia/.

[22] Brave, "Quiénes somos | grupo éxito." Disponible en https://www.grupoexito.com.co/es/quienes-somos.

[23] DANE, *Pobreza Monetaria en Colombia*. 2024.

[24] R Project, "R: The r project for statistical computing." Disponible en https://www.r-project.org/.

[25] nobledesktop, "How long does it take to learn r programming?." Disponible en https://www.nobledesktop.com/learn/r-programming/how-long-does-it-take-to-learn-r-programming.

[26] L. S. Vailshery, "Most used languages among software developers globally 2024." Disponible en https://www.statista.com/statistics/793628/worldwide-developer-survey-most-used-languages/.

[27] Plumber, "An api generator for r." Disponible en https://www.rplumber.io/.

[28] FastAPI, "Fastapi." Disponible en https://fastapi.tiangolo.com/.

[29] Zod, "Typescript-first schema validation with static type inference." Disponible en https://github.com/colinhacks/zod/.

[30] Next.js, "React foundations: About react and next.js | next.js." Disponible en https://nextjs.org/learn/react-foundations/what-is-react-and-nextjs.

[31] React, "React." Disponible en https://es.react.dev/.

[32] Aagam, "React hook form – a complete guide." Disponible en https://hygraph.com/blog/react-hook-form.

[33] E. D. P. Juristas, "Los riesgos legales del web scraping: Privacidad, protección de datos y malos usos." Disponible en https://www.edjxtechlawschool.com/post/los-riesgos-legales-del-web-scraping-privacidad-protecci~A{^3}n-de-datos-y-malos-usos.