

PONTIFICIA UNIVERSIDAD CATÓLICA DEL ECUADOR

FACULTAD DE INGENIERÍA

CARRERA DE: SISTEMAS DE INFORMACIÓN



TRABAJO DE TITULACIÓN

TEMA: APLICACIÓN MÓVIL PARA LA GESTIÓN DE NUTRICIÓN Y
DIETAS

ERICK CARIN MOLINA BASTIDAS

QUITO DM, JUNIO DEL 2024

DEDICATORIA

Este trabajo de titulación está dedicado a Dios, mis padres, abuelitos, hermanos, tíos, primos, los cuales han sido un apoyo incondicional en mi vida académica. En especial, a mi tío Daniel y mis abuelitos Fernando y Lupe, los cuales fueron las personas que siempre me apoyaron para salir adelante ante toda adversidad siempre dándome buenos consejos y ser mi inspiración cada día para superarme.

También dedico este trabajo a la persona más importante en mi vida, Fernanda, que ha sido un pilar fundamental siempre animándome y acompañándome durante este camino. Su amor, comprensión y motivación constante han sido esenciales para alcanzar este logro.

A mi hermana Naydelin y mi tía Gabriela, quienes estuvieron durante todo este proceso a mi lado, siempre alentándome y brindándome su apoyo.

Finalmente, dedico todo este esfuerzo realizado a quienes creyeron en mí y me ayudaron a mejorar cada día como persona y estudiante. Gracias a todas esas personas que considero un pilar fundamental en mi vida y ejemplo para seguir adelante.

AGRADECIMIENTO

Quiero expresar mis agradecimientos a todas las personas que han hecho posible y han contribuido con la realización de este trabajo de titulación. A mi director, por su conocimiento impartido a lo largo de este proceso, orientación y apoyo durante esta labor. Su experiencia y sabiduría han sido esenciales para el desarrollo de este trabajo de titulación. También, a todos los profesores que me han brindado herramientas y han aportado con su conocimiento durante este proceso.

RESUMEN

Este trabajo de titulación abarca el desarrollo de una aplicación móvil para la gestión de nutrición y generación de dietas de acuerdo a los objetivos y preferencias alimenticias del usuario, también muestra la ingesta calórica, de macronutrientes y agua consumidos a lo largo del día.

Esta aplicación busca ayudar a las personas a conseguir sus objetivos, los cuales pueden ser reducir de peso, mantenerlo o ganar masa muscular, teniendo en cuenta sus preferencias alimenticias, para ello la aplicación genera dietas para que el usuario pueda consumirlas de manera preferencial según el horario en que se encuentre, como puede ser desayuno, media mañana, almuerzo y merienda. De esta forma al momento de que el usuario registre una dieta consumida la aplicación mostrará las macronutrientes que se han consumido.

La aplicación fue desarrollada con herramientas como Flutter como Front-end y NestJs como Back-end, de la misma manera se utilizó MySQL como gestor de base de datos relacional y Firebase para el almacenamiento de imágenes. También, se utilizó Prototipado Evolutivo como marco metodológico para el desarrollo de la aplicación móvil.

ÍNDICE

Contenido

1. CAPÍTULO 1: INTRODUCCIÓN.....	12
1.1. TEMA.....	12
1.2. JUSTIFICACIÓN.....	12
1.3. PLANTEAMIENTO DEL PROBLEMA.....	13
1.4. OBJETIVOS.....	16
1.4.1. <i>Objetivo General</i>	16
1.4.2. <i>Objetivos Específicos</i>	16
1.5. ANTECEDENTES.....	17
1.6. ALCANCE.....	19
2. CAPÍTULO 2: FUNDAMENTACIÓN TEÓRICA.....	20
2.1. MARCO CONCEPTUAL.....	20
2.1.1. <i>Desarrollo Móvil Nativo</i>	20
2.1.2. <i>Desarrollo Móvil Híbrido</i>	20
2.1.3. <i>BackEnd</i>	20
2.1.4. <i>FrontEnd</i>	21
2.1.5. <i>Metodología</i>	21
2.1.6. <i>XP</i>	21
2.1.7. <i>Prototipado Evolutivo</i>	21
2.1.9. <i>API REST</i>	22
2.1.10. <i>Bases de Datos</i>	22
2.1.11. <i>Bases de Datos Relacional</i>	22
2.1.12. <i>Bases de Datos No Relacional</i>	23
2.1.13. <i>Diagrama Conceptual</i>	23
2.1.14. <i>Diagrama Físico</i>	23

2.1.15.	Diagrama Lógico	23
2.1.16.	SQLite.....	24
2.1.18.	Flutter.....	24
2.1.19.	IONIC.....	24
2.1.20.	React Native	24
2.1.21.	NestJS.....	25
2.2.	MARCO TEÓRICO	26
2.2.1.	Definiciones Fundamentales	26
2.2.1.1.	Desarrollo Móvil.....	26
2.2.1.2.	Nutrición.....	26
2.2.1.3.	Dieta Saludable	26
2.2.2.	Cuadros Comparativos de Herramientas	29
2.2.2.1.	Flutter, IONIC y React Native	29
2.2.2.2.	Bases de Datos Relacionales y No Relacionales.....	32
2.2.2.3.	Metodología XP y Prototipado Evolutivo	33
3.	CAPÍTULO 3: METODOLOGÍA	35
3.1.	METODOLOGÍA DE DESARROLLO IMPLEMENTADA	35
3.1.1.	Fases del Prototipado Evolutivo	36
3.1.1.1.	Recopilación de Requisitos.....	36
3.1.1.2.	Desarrollo del Prototipo Inicial.....	36
3.1.1.3.	Evaluación del Prototipo.....	37
3.1.1.4.	Refinamiento del Prototipo	37
3.1.1.5.	Iteración del Proceso.....	37
3.1.1.6.	Entrega del Producto Final.....	38
3.1.2.	Criterios de Evaluación.....	38
3.1.3.	Feedback de usuario	39
3.2.	PLANIFICACIÓN.....	39
4.	CAPÍTULO 4: DISEÑO.....	48

4.1.	PATRÓN DE DISEÑO	48
4.1.1.	<i>Componentes de MVVM</i>	49
4.2.	DISEÑO DE BASE DE DATOS.....	50
4.2.1.	<i>Diagrama Lógico</i>	50
5.	CAPÍTULO 5: DESARROLLO.....	52
5.1.	ESTÁNDARES EN EL DESARROLLO DE LA APLICACIÓN	52
5.2.	ORGANIZACIÓN DE MÓDULOS Y COMPONENTES.....	55
5.3.	LEVANTAMIENTO DE REQUERIMIENTOS	56
5.3.1.	<i>Historias de Usuario</i>	56
5.4.	PROTOTIPOS.....	64
5.4.1.	<i>Prototipo Gestión de Registro de Datos Personales</i>	64
5.4.1.1.	Desarrollo del Prototipo.....	64
5.4.1.3.	Feedback del Usuario.....	69
5.4.1.4.	Refinación del Prototipo	69
5.4.2.	<i>Prototipo Autenticación dentro de la Aplicación</i>	69
5.4.2.1.	Desarrollo del Prototipo.....	69
5.4.2.2.	Pruebas del Prototipo	70
5.4.2.3.	Feedback del Usuario.....	73
5.4.2.4.	Refinación de Prototipo	73
5.4.3.	<i>Prototipo Gestión de Dietas Personalizadas</i>	73
5.4.3.1.	Desarrollo del Prototipo.....	73
5.4.3.2.	Pruebas del Prototipo	74
5.4.3.3.	Feedback del Usuario.....	81
5.4.3.4.	Refinación del Prototipo	81
5.4.4.	<i>Prototipo Gestión de Ingesta de Agua</i>	81
5.4.4.1.	Desarrollo del Prototipo.....	81
5.4.4.2.	Feedback del Usuario.....	82
5.4.4.3.	Refinamiento del Prototipo	82

5.5.	MUESTRA DE LA APLICACIÓN	83
5.5.1.	<i>Registro dentro de la Aplicación</i>	83
5.5.2.	<i>Login de la Aplicación</i>	85
5.5.3.	<i>Actualización del Perfil de Usuario</i>	86
5.5.4.	<i>Elección de Objetivo y Preferencias</i>	87
5.5.5.	<i>Generación de Dietas Personalizadas</i>	88
5.5.6.	<i>Consumos de Dietas, Contador de Macronutrientes y Contador de Agua</i>	89
6.	CONCLUSIONES Y RECOMENDACIONES	91
6.1.	CONCLUSIONES:.....	91
6.2.	RECOMENDACIONES:	93
7.	ANEXOS	95
7.1.	ENTREVISTA PARA LEVANTAMIENTO DE REQUERIMIENTOS	95
8.	BIBLIOGRAFÍA.....	98

ÍNDICE DE FIGURAS

Figura 1	Indicador de salud sobre hábitos alimenticios en adultos ecuatorianos.....	14
Figura 2	Planificación de Actividades Semana 1	42
Figura 3	Planificación de Actividades Semana 2 y 3	43
Figura 4	Planificación de Actividades Semana 4 y 5	44
Figura 5	Planificación de Actividades Semana 6 y 7	45
Figura 6	Planificación de Actividades Semana 8 y 9	46
Figura 7	Planificación de Actividades Semana 10	47
Figura 8	Planificación de Actividades Semana 11 y 12	47
Figura 9	Model View ViewModel	48
Figura 10	Diagrama Lógico de la Base de Datos	51
Figura 11	Servicio de la Clase Alimento	52
Figura 12	Controlador de la Clase Alimento	53
Figura 13	Método listen para el puerto y la IP	54
Figura 14	Módulo Configuración de la Base de Datos	55
Figura 15	Método Post del Módulo de Registro	66
Figura 16	Método Put para el usuario	67
Figura 17	Prueba de Carga al Módulo de Registro	68
Figura 18	Método Post de Login del Sistema	71
Figura 19	Prueba de carga al Módulo Login	72
Figura 20	Método Post de Alimentos	75

Figura 21 Método Get de Alimentos	76
Figura 22 Método Post de Dietas	77
Figura 23 Método Get de Dietas según el objetivo	78
Figura 24 Prueba de Carga de Alimentos	79
Figura 25 Prueba de Carga de Dietas	80
Figura 26 Pantalla de Registro de la Aplicación	84
Figura 27 Pantalla de Login de la Aplicación	85
Figura 28 Pantalla de Actualización de Perfil de la Aplicación	86
Figura 29 Pantalla de Elección de Objetivos y Preferencias de la Aplicación.....	87
Figura 30 Pantalla de Generación de Dietas de la Aplicación.....	89
Figura 31 Pantalla de Consumo de Dietas y Contadores de la Aplicación.....	90

ÍNDICE DE TABLAS

Tabla 1. Tabla Rango de Kcal a consumir por Día	28
Tabla 2. Tabla comparación de herramientas para desarrollo móvil.	31
Tabla 3. Tabla comparación Bases de Datos Relacionales y No Relacionales.....	32
Tabla 4. Tabla comparación Metodología de Desarrollo.....	33
Tabla 5. Tabla Historias de Usuario y Módulos	56
Tabla 6. Historia de Usuario Registro de información del usuario	57
Tabla 7. Historia de Usuario Registro de Información Nutricional.....	58
Tabla 8. Historia de Usuario Edición de información general del usuario.....	59
Tabla 9. Historia de Usuario Login y Logout.....	60
Tabla 10. Historia de Usuario Presentación de dietas según los requerimientos	61
Tabla 11. Historia de Usuario Administración de ingesta calórica diaria.....	62
Tabla 12. Historia de Usuario Administración de ingesta de agua diaria.....	63

Comentado [SM1]: ¿Por qué tiene otro tipo de letra en el índice?

1. Capítulo 1: Introducción

1.1.Tema

Aplicación móvil para la gestión de nutrición y dietas.

1.2.Justificación

El acceso a una alimentación adecuada y personalizada es un desafío para muchas personas en Ecuador. Ya sea debido a limitaciones económicas o a la falta de acceso a servicios de nutrición, existe una necesidad latente de brindar soluciones que aborden estas barreras. En este contexto, el desarrollo de una aplicación móvil se presenta como una respuesta concreta y efectiva para atender estas necesidades.

Según la FAO (Organizaciones de las Naciones Unidas para la Alimentación y la Agricultura) el Ecuador tiene casos tanto de desnutrición como de sobrepeso, 6 de cada 10 adultos sufren de obesidad, esto tiene como consecuencia enfermedades tales como la diabetes, hipertensión y otras afecciones al corazón, de la misma manera se registran alrededor de 1.1 millones de personas padecen de inseguridad alimentaria.

La inseguridad alimentaria es el carecimiento del acceso a los alimentos que son nutritivos para un crecimiento y desarrollo normal para llevar una vida saludable, todos estos problemas se dan por la desinformación de las personas de cómo llevar una vida saludable con alimentos de su preferencia, también otros de los factores que se deben considerar es la pobreza urbana y rural.

Por todo ello, la aplicación móvil que se propone tiene como objetivo primordial satisfacer las demandas de las personas que buscan una dieta personalizada y tener el conocimiento de lo que se está consumiendo, adaptada a sus características individuales, como

peso, altura y necesidades específicas. A través de esta aplicación, los usuarios podrán acceder a planes de nutrición y recetas diseñadas específicamente para sus objetivos, ya sea mejorar su salud, perder peso, ganar masa muscular o simplemente diversificar su alimentación en casa.

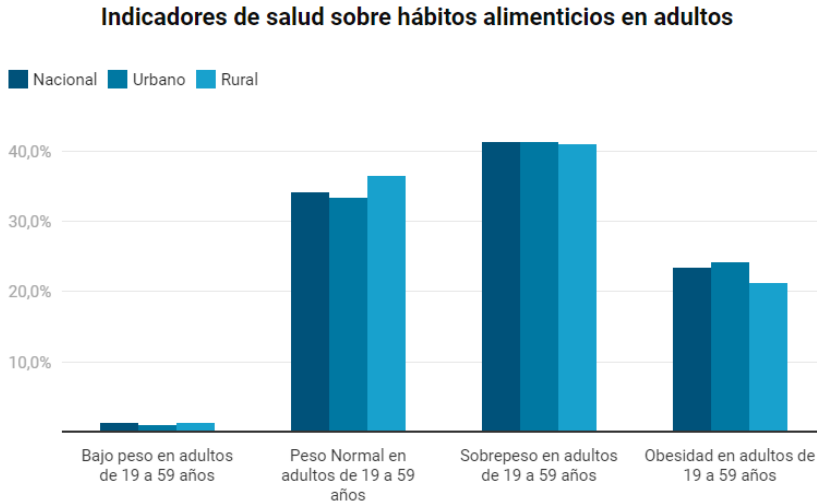
El enfoque principal de esta aplicación es proveer el acceso a una alimentación saludable y fomentar la adopción de hábitos alimenticios y nutrición adecuados en toda la población ecuatoriana. El diseño y desarrollo de la aplicación se basa en la premisa de que cualquier persona, sin importar su ubicación geográfica o situación económica, pueda acceder de manera sencilla y gratuita a planes de alimentación saludables y personalizados. Con solo abrir la aplicación en su dispositivo móvil, los usuarios encontrarán todo lo necesario para dar inicio a un camino hacia un estilo de vida más saludable.

1.3.Planteamiento del Problema

En la actualidad, muchas personas en Ecuador se enfrentan a la dificultad de adoptar un plan de alimentación saludable debido al alto costo de los servicios de nutrición. Como resultado, gran parte de la población ecuatoriana se encuentra atrapada en patrones de alimentación poco saludables, ya sea por falta de conocimiento o por la falta de acceso a orientación nutricional adecuada. Esta problemática afecta a numerosas familias en el país y, con el tiempo, puede llevar a problemas de salud.

Figura 1

Indicador de salud sobre hábitos alimenticios en adultos ecuatorianos.



Nota: La imagen representa las estadísticas sobre los hábitos alimenticios de los adultos ecuatorianos. Tomada de (Revista Gestión, 2020).

Este cuadro estadístico muestra de una manera detallada como las personas que están entre la edad de 19 a 59 años padecen de sobrepeso y obesidad es decir aproximadamente el 64% de la población no posee un peso de acuerdo a su altura, esto se presenta como consecuencia de la falta de conocimientos sobre los hábitos de alimentación e ingesta apropiada de alimentos y los macronutrientes necesarios de acuerdo a sus edad, peso y altura.

Por otro lado, algunas personas intentan llevar una alimentación más saludable, pero simplemente buscar recetas en Internet no es suficiente. Necesitan una guía personalizada que comprenda sus necesidades y especificaciones individuales para lograr sus objetivos de salud.

Además, la dependencia de las fuentes en línea para obtener planes de nutrición y dietas puede ser limitante, ya que estas fuentes a menudo ofrecen opciones genéricas y poco adaptadas a las necesidades individuales. Si alguien desea dietas más específicas, a menudo se ven obligados a pagar por servicios en línea, lo que puede ser un obstáculo para muchas personas en Ecuador que no desean gastar dinero en obtener una guía nutricional adecuada.

Para abordar esta problemática, se ha identificado la necesidad de crear una aplicación móvil que sea capaz de adaptarse a las especificaciones individuales de cada persona y proporcionar un plan de alimentación saludable. Esta aplicación tiene como objetivo empoderar a las personas para llevar una vida más saludable sin depender de costosos servicios de nutricionistas ni de soluciones limitadas en línea.

En función de esta problemática se plantea la siguiente pregunta principal de investigación:

¿Cómo se puede ayudar a las familias ecuatorianas a adoptar hábitos alimentarios más saludables mediante una aplicación móvil que gestione todas sus necesidades nutricionales de manera personalizada?

1.4.Objetivos

1.4.1. Objetivo General

Diseñar el prototipo de una aplicación móvil para la gestión de nutrición y dietas para las personas que estén interesadas en mejorar su calidad nutricional.

1.4.2. Objetivos Específicos

- Levantar toda la información relevante respecto al tema de nutrición y alimentación.
- Levantar toda la información relevante que permita generar las dietas considerando aspectos como peso, altura, edad, sexo, preferencias en alimentos y objetivo.
- Identificar las funciones esenciales de la aplicación, tales como el registro de datos, y generación de planes de dieta, con el fin de alcanzar el objetivo general de gestión de nutrición y dietas para cada uno de los usuarios de la aplicación.
- Construir un prototipo de aplicación móvil para la gestión de nutrición y dietas, asegurando la accesibilidad para todos los usuarios de las diferentes plataformas móviles.

1.5. Antecedentes

Hoy en día las personas buscan verse mejor consigo mismas, por ende, encuentran la manera de alimentarse de forma más saludable y verse bien, para ello la alimentación juega un rol muy importante dentro de su cambio, por lo consiguiente se pensó es una aplicación donde se pueda controlar la ingesta de alimentos para las personas que buscan mejorar su alimentación y tener un cambio.

Muchas personas no tienen conocimiento de cómo comer de manera más saludable de acuerdo a sus requerimientos, por ejemplo, existen muchas personas que desean bajar su índice de grasas, pero por otro lado hay personas que buscan aumentar de peso. Las personas desconocen cómo encontrar sus dietas adecuadas, así mismo encontrar planes y dietas que se ajusten a sus requerimientos, de la misma manera, en Ecuador una cita en clínicas especializadas en nutrición ronda entre los 60\$ a 75\$, mientras que una aplicación web cobra una suscripción mensual de 19.99\$ y anual de 99.99\$.

Por este tipo de problemas las personas no pueden cumplir sus requerimientos, está es la razón que con una aplicación móvil se desea solventar todo este tipo de situaciones, así las personas podrán encontrar dietas que se ajusten a ellos de acuerdo a su información de composición corporal.

Como soporte de este trabajo de titulación existen tesis que pueden ayudar a mejorar la calidad de la aplicación y elegir las mejores herramientas, e incluso mejorarlas, con estas se puede obtener información que pueda contribuir al caso de estudio presente, se han tomado de referencia 2 tesis las cuales será de ayuda con los conocimientos que contienen, las mismas

portan información sobre las bases de datos, conocimientos sobre el área de nutrición que esta puede ser muy importante para no pasar en alto investigaciones que puedan ser valiosas.

Se tiene como primera fuente de información la tesis de realizada por Cargua Jiménez & Fernández Hidalgo (2017) con el tema “Desarrollo de una aplicación móvil para el control nutricional y desarrollo infantil como una forma de prevenir la obesidad” tiene como uno de sus objetivos específicos la recomendación de un plan nutricional dependiendo el diagnóstico del paciente, esta tesis proporciona fundamentos e información relevante al momento de sugerir y recomendar un plan nutricional a los pacientes en base a la información obtenida, lo cual esta tesis puede ser de ayuda para el desarrollo de la aplicación reutilizando y mejorando la forma de sugerir planes nutricionales.

De la misma manera la tesis de Santacruz Ayala & Villegas Proaño (2020) sobre “Desarrollo de aplicación móvil para planificación de dietas saludables” ayudará a determinar los requerimientos de la aplicación, este módulo puede ser de gran ayuda para la implementación de todos los requerimientos que se puedan pasar por alto, de la misma manera se pueden reutilizar estos requerimientos y mejorarlos para tener una aplicación mucho más completa.

Por otro lado, estas tesis tienen temas y desarrollo de las mismas interesantes, pero lo que se propone con el presente trabajo de titulación es darle más relevancia a la generación de dietas y generando un plus con lo que está consumiendo el usuario y la visualización de macronutrientes que se consumen, es decir al momento de elegir una dieta en el cual se esté interesado se registra todos los macronutrientes que el usuario está ingiriendo, todo esto con el fin de que se pueda seguir de manera controlada la alimentación de los usuarios.

1.6.Alcance

El presente trabajo de titulación tiene como resultado final el diseño de una aplicación móvil sobre la gestión de nutrición y dietas, el cual será elaborada en el framework Flutter que es propio del lenguaje de programación Dart, se ayudará también de otras herramientas las cuales son NestJS, MySql, JS, Cloud Storage, Jwt Token Authentication, Bcrypt , estas ayudarán a facilitar el desarrollo de la aplicación móvil, este desarrollo se llevará a cabo con la metodología de trabajo Prototipado Evolutivo que es un marco de trabajo ágil y adecuado para el desarrollo, por la forma de trabajar en cada iteración, lo cual permitirá mejorar los prototipos de cada módulo que se tengan en consideración a trabajar.

Se tendrá como resultado final dietas personalizadas para cada usuario según los requerimientos que él desee como pueden ser ganar peso, perder peso o simplemente mantener su peso corporal, dichas dietas tendrán un total de calorías por dieta y serán 4 comidas las que ayuden al usuario a completar su meta, por lo cual también se ha considerado que cada plato tendrá un porcentaje de proteínas, grasas y carbohidratos considerando que cada uno de ellos variará según el requerimiento que se haya elegido.

2. Capítulo 2: Fundamentación Teórica

2.1.Marco Conceptual

2.1.1. Desarrollo Móvil Nativo

El desarrollo móvil nativo describe la creación de una aplicación específicamente para un sistema operativo móvil determinado, como iOS o Android, utilizando lenguajes de programación nativos de ese sistema operativo. La creación de aplicaciones nativas genera una experiencia de usuario más fluida porque se comportan como otras aplicaciones nativas, funcionando mejor y permitiendo acceso directo a otras capacidades del dispositivo, como la cámara, el micrófono, el GPS, etcétera. (Fis, 2023)

2.1.2. Desarrollo Móvil Híbrido

Las aplicaciones híbridas representan una forma única de diseñar software para teléfonos inteligentes. Son una mezcla entre un sitio web y una aplicación nativa para móviles que ofrece al usuario la mejor experiencia posible sin comprometer la seguridad ni el rendimiento. Estas aplicaciones se pueden ejecutar tanto en iOS como en Android, lo que significa que los usuarios pueden acceder a sus datos desde diferentes dispositivos sin problemas. Por eso, además de ofrecer una solución práctica para los usuarios, las aplicaciones híbridas también permiten a los desarrolladores crear productos más rápidamente. (López, 2023)

2.1.3. BackEnd

El Backend es un término que utilizamos para referirnos a la arquitectura interna de un sitio web. Esta área lógica, que no es visible a los ojos del usuario y no incluye elementos de

tipo gráfico, permite que todos los elementos de una web desarrollen la función correcta. (Gómez, 2022)

2.1.4. FrontEnd

El Frontend es la parte del desarrollo web que se dedica a la parte frontal de un sitio web, en pocas palabras del diseño de un sitio web, desde la estructura del sitio hasta los estilos como colores, fondos, tamaños hasta llegar a las animaciones y efectos. (Bautista García, 2021)

2.1.5. Metodología

La metodología es la ciencia (o técnica) que utilizamos para realizar esa indagación de manera eficaz y conseguir los resultados deseados, con ella obtenemos la estrategia a seguir en el proceso de investigación. (Software DELSOL, 2019)

2.1.6. XP

La programación extrema es un método de desarrollo de software dividido en sprints de trabajo. Los marcos ágiles siguen un proceso iterativo, en el que se completa y revisa el marco al final de cada sprint, refinándolo para adaptarlo a los requisitos cambiantes y alcanzar la eficiencia máxima. (Raeburn, 2024)

2.1.7. Prototipado Evolutivo

Un modelo prototipo o modelo de desarrollo evolutivo es utilizado principalmente en el desarrollo de software para ofrecer al usuario una visión previa de cómo será el programa o sistema. Se le dice de desarrollo evolutivo al modelo de prototipo porque evoluciona hasta convertirse en el producto final. (Hosting Plus, 2021)

2.1.8. SCRUM

La metodología Scrum es un proceso para llevar a cabo un conjunto de tareas de forma regular con el objetivo principal de trabajar de manera colaborativa, es decir, para fomentar el trabajo en equipo. (APD, 2024)

2.1.9. API REST

REST significa transferencia de estado representacional. REST define un conjunto de funciones como GET, PUT, DELETE, etc. que los clientes pueden utilizar para acceder a los datos del servidor. Los clientes y los servidores intercambian datos mediante HTTP. (Amazon Web Services, Inc., s. f.-c)

2.1.10. Bases de Datos

Una base de datos es una recopilación de datos sistemática y almacenada electrónicamente. Puede contener cualquier tipo de datos, incluidos palabras, números, imágenes, vídeos y archivos. Puede usar un software denominado sistema de administración de bases de datos (DBMS) para almacenar, recuperar y editar datos. (Amazon Web Services, Inc., s. f.)

2.1.11. Bases de Datos Relacional

Una base de datos relacional es un tipo de base de datos que almacena y proporciona acceso a puntos de datos relacionados entre sí. Las bases de datos relacionales se basan en el modelo relacional, una forma intuitiva y directa de representar datos en tablas. En una base de datos relacional, cada fila en una tabla es un registro con una ID única, llamada clave. Las columnas de la tabla contienen los atributos de los datos y cada registro suele tener un valor

para cada atributo, lo que simplifica la creación de relaciones entre los puntos de datos.

(Oracle, s. f.-b)

2.1.12. Bases de Datos No Relacional

Las bases de datos NoSQL están diseñadas específicamente para modelos de datos específicos y almacenan los datos en esquemas flexibles que se escalan con facilidad para aplicaciones modernas. Las bases de datos NoSQL son ampliamente reconocidas porque son fáciles de desarrollar, por su funcionalidad y el rendimiento a escala. Esta página incluye recursos que lo ayudan a comprender mejor las bases de datos NoSQL y comenzar a usarlas.

(Amazon Web Services, Inc., s. f.-a)

2.1.13. Diagrama Conceptual

Un modelo conceptual de datos identifica las relaciones de más alto nivel entre las diferentes entidades. (Tecnologías Información, s. f.)

2.1.14. Diagrama Físico

El modelo de datos físicos representa cómo se construirá el modelo en la base de datos. Un modelo de base de datos física muestra todas las estructuras de tabla, incluidos el nombre de columna, el tipo de datos de columna, las restricciones de columna, la clave principal, la clave externa y las relaciones entre las tablas. (Tecnologías Información, s. f.)

2.1.15. Diagrama Lógico

Un modelo de datos lógicos describe los datos con el mayor detalle posible, independientemente de cómo se implementarán físicamente en la base de datos. (Tecnologías Información, s. f.)

2.1.16. SQLite

SQLite es una herramienta de software libre, que permite almacenar información en dispositivos empotrados de una forma sencilla, eficaz, potente, rápida y en equipos con pocas capacidades de hardware, como puede ser una PDA o un teléfono celular. (Rommel, s. f.)

2.1.17. MySQL

MySQL es la base de datos de código abierto más popular del mercado. Según DB-Engines, MySQL se clasifica como la segunda base de datos más popular, detrás de Oracle Database. MySQL potencia muchas de las aplicaciones más accesibles, como Facebook, Twitter, Netflix, Uber, Airbnb, Shopify y Booking.com. (Oracle, s. f.-a)

2.1.18. Flutter

Flutter es un marco de código abierto desarrollado y compatible con Google. Los desarrolladores de front-end y pila completa utilizan Flutter para crear una interfaz de usuario (IU) de aplicación para varias plataformas con un único código base. (Amazon Web Services, Inc., s. f.-b)

2.1.19. IONIC

Ionic Framework es un SDK de frontend de código abierto para desarrollar aplicaciones híbridas basado en tecnologías web (HTML, CSS y JS). Es decir, un framework que nos permite desarrollar aplicaciones para iOS nativo, Android y la web, desde una única base de código. (Atmitim & Maluenda, 2021)

2.1.20. React Native

React Native es un framework JavaScript para crear aplicaciones reales nativas para iOS y Android, basado en la librería de JavaScript React para la creación de componentes visuales, cambiando el propósito de los mismos para, en lugar de ser ejecutados en navegador, correr directamente sobre las plataformas móviles nativas, en este caso iOS y Android. (Deloitte Spain, s. f.)

2.1.21. NestJS

Nest.js es uno de los frameworks de Node.js de más rápido crecimiento para construir aplicaciones backend eficientes, escalables y de nivel empresarial utilizando Node.js. Es conocido por producir aplicaciones altamente comprobables, mantenibles y escalables utilizando el moderno JavaScript y TypeScript. (Kinsta, 2022)

2.2.Marco Teórico

2.2.1. Definiciones Fundamentales

2.2.1.1. Desarrollo Móvil

El desarrollo de aplicaciones móviles es el proceso de creación de software para smartphones, tablets y asistentes digitales, más comúnmente para los sistemas operativos Android y iOS. El software se puede preinstalar en el dispositivo, descargar desde una tienda de aplicaciones móviles o acceder a través de un navegador web móvil. Los lenguajes de programación y marcación utilizados para este tipo de desarrollo de software incluyen Java, Swift, C# y HTML5. (IBM, s. f.)

2.2.1.2. Nutrición

Nutrición consiste en comer una variedad de alimentos, limitar el consumo de ciertos alimentos y bebidas y controlar la cantidad de alimentos y calorías que se ingieren. Una alimentación equilibrada ayuda a reducir el riesgo cardiovascular porque reduce tanto el colesterol y la presión arterial como el peso. (The Texas Heart Institute, 2019)

2.2.1.3. 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. (Organización Mundial de la Salud, 2019)

Una dieta considerada saludable incluye una alta ingesta de frutas, una ingesta moderada de legumbres, pescado, cereales integrales y aceites vegetales, por el contrario, una ingesta reducida de cereales refinados y carnes rojas (particularmente carnes procesadas con sodio), el azúcar, la sal y las grasas trans se encuentran en los alimentos industriales tales como la margarina, los productos de panadería y comidas rápidas.

Es importante enfatizar que una dieta subóptima es una de las principales causas de mala salud, hoy en día reconocemos el impacto que tanto la dieta como los hábitos alimentarios tienen en la salud a largo plazo. Destaca la importancia de una alimentación saludable. Esto incluye consumir macronutrientes (proteínas, carbohidratos y grasas) en proporciones adecuadas para satisfacer las necesidades energéticas y fisiológicas sin exceder la ingesta.

También se debe considerar que una hidratación adecuada y un aporte suficiente de micronutrientes (vitaminas y minerales), estos son esenciales para el crecimiento, el desarrollo, el metabolismo y la función fisiológica normal, incluso si se necesitan cantidades menores que los macronutrientes.

2.2.1.4. Calorías Necesarias

Para una buena personalización de las dietas según el usuario se debe tener en cuenta la ingesta calórica de la persona, por ejemplo:

Rango Kcal/Día	Perder Peso	Ganar Masa	Mantener Peso
		Muscular	
Hombres	1500-2200	2500-3500	2200-3000
Mujeres	1200-1800	2200-3250	1800-2500

Tabla 1. Tabla Rango de Kcal a consumir por Día

Autor. Realizada por el Autor

En base a esta información se podrá tomar en cuenta cuales son los parámetros para poder dar una dieta adecuada según el rango calórico que se tiene, así mismo para poder llegar a las calorías necesarias se deberá tomar en cuenta los macronutrientes y la ingesta de los mismos.

2.2.2. Cuadros Comparativos de Herramientas

2.2.2.1. Flutter, IONIC y React Native

El cuadro comparativo de estas tecnologías de desarrollo está basado en la presente demanda que se tiene en el mercado al momento de utilizar estas herramientas, así como son su forma de uso, portabilidad y escalabilidad.

	Flutter	IONIC	React Native
Diseño de Interfaces	Diseño más profesional utilizando componentes UI.	Diseños de interfaces sencillos y fácil de aprender.	Componentes mucho más uniformes de acuerdo al dispositivo.
Aprendizaje	Sencillo a pesar de ser Framework de Dart.	Sencillo de aprender, se basa en HTML, CSS y JS.	Se debe tener conocimientos previos de React (Framework de JS).
Tiempo de Desarrollo	El tiempo de desarrollo es mucho menor al emplear características en menos tiempo.	Su tiempo de desarrollo es una de la más rápidas ya que no requiere necesariamente de la aplicación de una de las tecnologías actuales.	Simplifica el tiempo de desarrollo aprovechando los componentes listos para poder crear funciones.

Funciones	Puede presentar ciertos problemas en la utilización de algunas funciones en IOS debido a su tiempo de actualización dentro de esta plataforma.	En ciertas ocasiones se necesitan plugins para poder implementar ciertas funciones.	Para adaptarse a ciertas funciones que están dentro del mercado actualmente aún se necesita cierto soporte de algunas funciones.
Tamaño de la aplicación	Ciertas aplicaciones por el kit de desarrollo que se pueden utilizar al momento de desarrollar pueden presentar problemas al tamaño de la aplicación.	Al momento de utilizar HTML, CSS y JS, junto con dependencias y si se utilizan Frameworks tienden a ser aplicaciones más pesadas.	Sus aplicaciones tienden a ser pesadas, adicionando que React no sabe cómo distribuir de manera adecuada el uso del hardware de manera que pueda que la app no corra de manera óptima.
Actualización de Código dentro del entorno de desarrollo.	Tiene función de Hot Reloading para cambios dentro de la interfaz.	Tiene la función Live Reloading para cambios rápidos dentro de la interfaz.	Tiene función de Hot Reloading para cambios dentro de la interfaz.

Integración con el BackEnd	Posee una fácil integración con API REST ya que posee el paquete 'https' el cual permite traer los métodos post, put, get, delete de una manera sencilla con los parámetros de sus funciones.	Posee una fácil integración de API REST mediante Axios y Fetch API	Posee una fácil integración con REST API mediante Axios y Fetch API
Desventajas	El tamaño de la aplicación puede llegar a ser muy grande.	El rendimiento no puede ser tan fluido en comparación de Flutter o React Native.	Su desarrollo puede llegar a complicarse debido a la integración de algunos módulos.

Tabla 2. Tabla comparación de herramientas para desarrollo móvil.

Autor. Elaborado por el Autor

Ante este análisis mediante un cuadro comparativo se ha llegado a la conclusión que una de las herramientas que me podrá ayudar a realizar la app de manera adecuada tomando en cuenta los aspectos de diseño de interfaces, curva de aprendizaje y rendimiento de los componentes es Flutter a pesar de que pueda tener

desventajas es una de las opciones más adecuadas para poder implementar y desarrollar la aplicación.

2.2.2.2. Bases de Datos Relacionales y No Relacionales

Una parte muy importante al momento de construir un sistema o una aplicación móvil es que tipo de Bases de Datos utilizar si una relacional o no relacional, pues eso dependerá de la aplicación y con qué fin se lo realiza, pero aquí uno de los puntos importantes a considerar cada uno de ellos.

	Relacionales	No Relacionales
Identificadores	Es obligatorio el uso de identificadores entre tablas.	No es necesario el uso de un identificador para sus tablas.
Estructura	Debe tener una conexión con otras tablas, es decir debe tener dependencia con 1 o más tablas.	No necesita tener dependencia entre otras tablas.
Tipo de Desarrollo	Se utiliza para el desarrollo de sistemas contables, facturación, inventario, etc.	Su uso es fundamental para la creación de redes sociales.

Tabla 3. Tabla comparación Bases de Datos Relacionales y No Relacionales.

Autor. Elaborado por el Autor

2.2.2.3. Metodología XP y Prototipado Evolutivo

	XP	Prototipado Evolutivo
Participación del Cliente	El cliente se encuentra en todas las etapas del desarrollo.	El cliente está dentro de la evaluación y retroalimentación de los prototipos.
Flexibilidad	Se adapta a los cambios según los requisitos del cliente.	Ajusta y mejora los prototipos según la retroalimentación del cliente.
Ciclo de Desarrollo	Entregas frecuentes y pequeñas, todas en base a su sprint.	Iterativo y enfocado en la creación de los prototipos.
Documentación	Debe ser lo suficientemente consistente al momento del desarrollo y el mantenimiento después del mismo.	Incrementa conforme el prototipo sigue avanzando.
Aplicación en el desarrollo	Es una buena metodología para la calidad y puede implementar cambios de una manera rápida.	Ajustar la aplicación de acuerdo a la retroalimentación del usuario.

Tabla 4. Tabla comparación Metodología de Desarrollo.

Autor. Elaborado por el Autor

La metodología para empezar el desarrollo de una aplicación es muy importante ya que de acuerdo a su metodología se entregarán prototipos o pequeños sprints los cuales serán muy fundamentales para el cliente, ya que él debe estar involucrado para ver cómo queda el producto final. Para este caso se ha elegido la Metodología de Prototipado Evolutivo, la cual es una de las mejores de acuerdo a mis requerimientos, sobre todo por su flexibilidad y por el compromiso que debe tener el cliente dentro del desarrollo de la aplicación que en este caso será la persona encargada de revisar temas de nutrición.

3. Capítulo 3: Metodología

3.1. Metodología de Desarrollo Implementada

Dentro del presente trabajo se pudo comparar 2 metodologías y las características que cada una de estas presentaba, la Tabla 4 presenta esta información para lo cual se ha optado por utilizar la metodología de Prototipado Evolutivo, la cual permite el desarrollo de prototipos para cada módulo de la aplicación. Además, trabaja en manera iterativa, siempre retroalimentado cada uno de los prototipos desarrollados en función a las necesidades del usuario

La ventaja de esta metodología es que se podrá trabajar de la mano con la parte interesada, en este caso cada final de la iteración se probará el módulo y se dará un resumen del mismo, así también se irá probando la funcionalidad de cada uno de estos, si de ser el caso se necesita una corrección otra vez se ejecutará la iteración, actuando en forma de espiral hasta pasar al próximo módulo o prototipo.

Dentro del trabajo con el usuario se tendrán reuniones de manera virtual para que se pueda presentar los prototipos trabajados, así mismo recoger una retroalimentación en el caso de que sea necesario en base a un feedback que se presentará mediante una encuesta cuando se finalice cada módulo, esta parte será muy fundamental debido a que se deberán ajustar con los requisitos que el usuario esté solicitando, una vez que esté satisfecho se procederá con la siguiente iteración, hasta conseguir el producto final deseado.

3.1.1. Fases del Prototipado Evolutivo

3.1.1.1. Recopilación de Requisitos

Dentro de esta fase se recopilan todos los requisitos que sean necesarios para llevar a cabo el desarrollo de la aplicación, en este caso la información será aportada por una persona del área de la nutrición, la cual aportará ideas, información sobre los requisitos que se necesitan para poder realizar la aplicación.

Para el levantamiento de requerimientos se hará una entrevista personal en donde se explicará la idea de la aplicación y se recopilarán los datos más importantes que se deberían tomar en cuenta, también en base a la información recolectada se podrá determinar los módulos que se trabajarán dentro de ella y poder obtener información sobre el área nutricional e información que sea relevante para el desarrollo del mismo.

3.1.1.2. Desarrollo del Prototipo Inicial

La fase de desarrollo del prototipo inicial se lleva a cabo construyendo un prototipo con los requisitos recogidos para cada iteración, el desarrollo solo es una versión sencilla del sistema, únicamente para que el usuario pueda tener una primera visión de cómo será el aplicativo.

El desarrollo del prototipo inicial se llevará a cabo con las funcionalidades que se consideraron dentro de la entrevista, así mismo se tendrá un diseño rápido de este prototipo con el fin de que el usuario pueda ofrecer una retroalimentación de cómo podría ser un diseño referente al área de la nutrición.

3.1.1.3. Evaluación del Prototipo

En este punto el usuario probará el prototipo y proporcionará una retroalimentación del mismo, para esta fase el usuario podrá identificar errores dentro de su funcionalidad y podrá ofrecer mejoras para el prototipo en caso de ser necesario.

Se llevarán a cabo actividades clave, como son las pruebas que se realice a la aplicación y la recolección del feedback del usuario. Además, se analizarán los resultados de las pruebas que se realicen a la aplicación y se utilizará la retroalimentación que se obtenga por parte del usuario para mejorar el prototipo antes de avanzar con la siguiente iteración.

3.1.1.4. Refinamiento del Prototipo

En base a la retroalimentación recibida por el usuario, en esta fase se modificará y se mejorará el prototipo, esta fase puede repetirse ajustando el prototipo hasta que se cumpla con las expectativas del usuario.

Las actividades que se llevarán a cabo serán las correcciones de errores que pueda tener el prototipo, también se podrán adicionar funcionalidades que puedan llegar a faltar y la reestructuración del diseño en el caso de que el usuario lo solicite.

3.1.1.5. Iteración del Proceso

Las fases mencionadas anteriormente se deben repetir iterativamente de ser necesario, todo esto con el fin de que el prototipo sea más completo y sin fallas, con el fin de tener el prototipo final funcional.

Las actividades que se llevarán a cabo aquí serán el refinamiento del prototipo constantemente y la retroalimentación continua por parte del usuario.

3.1.1.6. Entrega del Producto Final

En esta fase una vez que se haya alcanzado un estado satisfactorio por parte del usuario y cumplido con todos los requisitos que se han solicitado pasará a convertirse en el producto final.

Las actividades a llevarse a cabo dentro de esta fase son pruebas finales del producto final, así como la aprobación del usuario, y se asegurará de que todo esté de acuerdo a los requisitos solicitados, también se aplicarán ajustes en caso de que sean necesarios.

3.1.2. Criterios de Evaluación

Se evaluará cada iteración de los prototipos en base a las pruebas que se realicen dentro de cada módulo, los cuales son:

- **Integración:** Se realizarán pruebas de integración para que se pueda comprobar que cada uno de los módulos se integren de la manera correcta con la importación y exportación de datos desde la BD, esta prueba se la ejecutará en los métodos API REST, para ello se utilizará la herramienta ThunderClient (Visual Studio Code).
- **Carga:** Se aplicarán pruebas de carga a los métodos API REST que estén presentes dentro de la aplicación, las pruebas se realizarán con la herramienta Artillery (NestJs), la cual nos permitirá ver las estadísticas de carga que se

aplican dentro de un tiempo y usuarios determinados, dando como resultado la cantidad de usuarios que pueden utilizar los métodos del sistema al mismo tiempo.

3.1.3. Feedback de usuario

- **Análisis de uso:** Se realizarán encuestas de satisfacción para cada prototipo del módulo que se presente al usuario y reuniones virtuales para poder recolectar su retroalimentación.

3.2. Planificación

Dentro de esta fase se describirá de qué forma se irá definiendo como se desarrollará el aplicativo, en las cuales se levantaron los requerimientos funcionales de toda la aplicación, para ello se aplicó una entrevista a una persona dentro del área de nutrición, con la que se pudo entender cada una de las definiciones necesarias, así como los datos que se necesitan para procesar las dietas.

El objetivo fue tener en claro cómo funcionará cada uno de los módulos y entender cómo estará relacionada la base de datos de acuerdo a la información que se ha obtenido de la entrevista, por ende, los módulos que se han tenido en cuenta son los siguientes:

- **Gestión de Registro de datos personales:** Para este módulo se considerará la recolección de los datos generales del usuario, todo esto con el fin de que se almacenen para tener en cuenta los usuarios registrados para el siguiente módulo de autenticación.

- **Autenticación dentro de la aplicación:** Dentro de este módulo se gestionará el login y logout dentro del sistema, asegurando así la encriptación de datos sensibles como la clave del usuario y el token de sesión, logrando una autenticación segura dentro la aplicación.
- **Gestión de dietas personalizadas:** El módulo de gestión de dietas permitirá al usuario elegir entre 3 objetivos los cuales son: Ganar Masa Muscular, Perder Peso, Mantener Peso, para posteriormente elegir los alimentos que el usuario prefiera, después se generarán dietas en base a su objetivo y preferencias alimenticias. Una vez generada las dietas se presentarán en una interfaz todas las dietas que se han generado para cada horario de comida, de la misma manera el usuario podrá ver las dietas a más detalle como sus ingredientes, tiempo de preparación y macronutrientes que contiene esa dieta. También, se presentará una interfaz en donde el usuario podrá visualizar la cantidad de macronutrientes que está ingiriendo en base a las dietas seleccionadas de acuerdo a la hora del día.
- **Gestión de ingesta diaria de agua:** El módulo de ingesta de agua se presentará de manera que el usuario pueda registrar la cantidad de agua que está consumiendo a lo largo del día, la ingesta de agua irá de acuerdo a la información que se ha presentado en la fase de levantamiento de requerimientos.

Estos módulos se tomaron en cuenta debido a la encuesta realizada a la persona entendida del área de nutrición, la cual prestó su conocimiento para los requerimientos que

debería tener la aplicación como lo son: El entendimiento de cada uno de los objetivos de las personas y entre que rango se consideran las calorías como objetivo, también la importancia que tiene la ingesta de agua para la parte nutricional de la persona, así mismo cada persona debe poder elegir que alimentos ingerir, ya que pueden existir alergias por parte de la persona.

Se consideraron importantes los datos del registro del usuario, así como los datos que se debían tomar en cuenta para posteriormente evaluarlos. Esto datos fueron el peso, género, talla, edad, ya que, en base a ellos, dentro de la aplicación se considera importante el cálculo diario para la ingesta de macronutrientes de la persona. Esta información permitirá asegurar que el usuario solo consumir las calorías necesarias en base al objetivo que han elegido.

Por otro lado, dentro de esta fase de planificación se ha elaborado un calendario en donde se explica de manera detallada las semanas en las cuales se realizarán cada una de las actividades, desde el levantamiento de requerimientos, presentación de cada prototipo de los módulos, retroalimentaciones presentadas de ser el caso, afinamiento de los prototipos, pruebas de la aplicación y revisión final de la aplicación.

La Figura 2 representa las actividades que se trabajarán dentro de la semana 1, las cuales son el levantamiento de requerimientos, así mismo la revisión y la afinación que se darán a los requerimientos recogidos, cada una de estas actividades tendrán una prioridad alta para el sistema.

Figura 2

Planificación de Actividades Semana 1

▼ **Semana 1**

<input type="checkbox"/>	Proyecto		Responsable	Estado	Prioridad
<input type="checkbox"/>	Levantamiento de Requ...	+		Listo	Alta
<input type="checkbox"/>	Revisión de Requerimie...	+		Listo	Alta
<input type="checkbox"/>	Afinación de Requerimi...	+		Listo	Alta
<input type="checkbox"/>	+ Agregar proyecto				




Autor. Elaborado por el Autor

La Figura 3 indica las actividades que se trabajarán en la semana 2 y 3, en estas semanas se desarrollará el prototipo del módulo de registro, así mismo la evaluación del prototipo, la cual está compuesta por las pruebas que se realizarán y la retroalimentación que pueda existir por parte del usuario, de igual forma se manejará el refinamiento del prototipo.

Figura 3

Planificación de Actividades Semana 2 y 3

✓ **Semana 2/3**

<input type="checkbox"/>	Proyecto		Responsable	Estado	Prioridad
<input type="checkbox"/>	Desarrollo del Primer Pr...	+		Listo	Alta
<input type="checkbox"/>	Evaluación de Prototipo	+		Listo	Alta
<input type="checkbox"/>	Refinamiento del Protot...	+		Listo	Alta
<input type="checkbox"/>	+ Agregar proyecto				

Autor. Elaborado por el Autor

La **Figura 4** representa las actividades a realizarse dentro de la semana 4 y 5, se realizará el desarrollo del segundo prototipo el cual consiste en la autenticación dentro de la aplicación, esta se basará en el login y logout de la sesión del usuario, de la misma manera se realizarán pruebas y se obtendrá feedback del usuario, así como el refinamiento del sistema en base a la retroalimentación que pueda existir.

Figura 4

Planificación de Actividades Semana 4 y 5

▼ Semana 4/5

<input type="checkbox"/>	Proyecto		Responsable	Estado	Prioridad
<input type="checkbox"/>	Desarrollo del Segundo ...	+		Listo	Alta
<input type="checkbox"/>	Evaluación del Prototipo	+		Listo	Alta
<input type="checkbox"/>	Refinamiento del Protot...	+		Listo	Alta
<input type="checkbox"/>	+ Agregar proyecto				






Autor. Elaborado por el Autor

La **Figura 5** representa las actividades de la semana 6 y 7, en donde se elaborará el tercer módulo, el cual consiste en la gestión de dietas personalizadas, aquí se trabajará en la generación de dietas, muestra de las dietas a los usuarios y la ingesta de cada dieta, así como el contador de macronutrientes consumidos, en este módulo también se realizarán pruebas y recolectará la retroalimentación, de igual manera se aplicará un refinamiento del prototipo en caso de ser necesario.

Figura 5

Planificación de Actividades Semana 6 y 7

▼ **Semana 6/7**

<input type="checkbox"/>	Proyecto		Responsable	Estado	Prioridad
<input type="checkbox"/>	Desarrollo del Tercer Pr...	+		Listo	Alta
<input type="checkbox"/>	Evaluación de Prototipo	+		Listo	Alta
<input type="checkbox"/>	Refinamiento del Protot...	+		Listo	Alta
<input type="checkbox"/>	+ Agregar proyecto				
					

Autor. Elaborado por el Autor

La Figura 6 representa las actividades de las semanas 8 y 9, en la cual se realizará el desarrollo del módulo de la ingesta de agua, para el cual se elaborará la recolección de feedback del usuario, por otra parte, no se aplicarán pruebas, ya que al tener pruebas de integración y carga para la parte de Back-end, las pruebas no aplicarán dentro de este módulo.

Figura 6

Planificación de Actividades Semana 8 y 9

▼ **Semana 8/9** 3 Proyectos

<input type="checkbox"/>	Proyecto		Responsable	Estado	Prioridad
<input type="checkbox"/>	Desarrollo del Cuarto Pr...	+		Listo	Alta
<input type="checkbox"/>	Evaluación de Prototipo	+		Listo	Alta
<input type="checkbox"/>	Refinamiento del Protot...	+		Listo	Alta
<input type="checkbox"/>	+ Agregar proyecto				

Autor. Elaborado por el Autor

La Figura 7 se muestra la actividad de la semana 10, se realizará la prueba de sistema de toda la aplicación, también se consideró dar una prioridad alta como todas las actividades que se trabajarán en el calendario establecido.

Figura 7

Planificación de Actividades Semana 10

▼ **Semana 10**

<input type="checkbox"/>	Proyecto		Responsable	Estado	Prioridad
<input type="checkbox"/>	Pruebas de la Aplicación	+		Listo	Alta
<input type="checkbox"/>	+ Agregar proyecto				

Autor. Elaborado por el Autor

La Figura 8 muestra las actividades de la semana 11 y 12, en donde se llevarán a cabo la revisión final de la aplicación y en caso de que haya una retroalimentación adicional por parte del usuario realizar el refinamiento, caso contrario, se daría por culminado el producto.

Figura 8

Planificación de Actividades Semana 11 y 12

▼ **Semana 11/12**

<input type="checkbox"/>	Proyecto		Responsable	Estado	Prioridad
<input type="checkbox"/>	Revisión Final de la Apli...	+		Listo	Alta
<input type="checkbox"/>	Refinamiento de Revisi...	+		Listo	Alta
<input type="checkbox"/>	+ Agregar proyecto				

Autor. Elaborado por el Autor

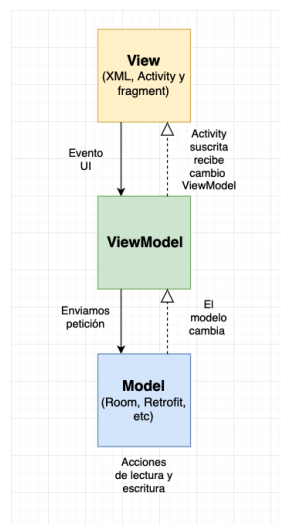
4. Capítulo 4: Diseño

4.1. Patrón de diseño

La Figura 9 representa el patrón de diseño de la aplicación, la cual se desarrollará mediante la MVVM (Model View-View Model) que trabaja en capas al igual que otros patrones de diseño, lo que facilita es que se no se puede saltar entre capas, todo esto para poder tener mayor seguridad en el transporte de los datos, al igual que integridad de los mismos, este consiste en traer los datos desde el backend (NestJs) hasta el frontend (Flutter) y desde allí trabajarlo en este modelo basado en capas.

Figura 9

Model View ViewModel



Nota: La imagen de representa como trabaja entre capas el patrón de diseño MVVM. Tomada de (Kotlin, 2021)

4.1.1. Componentes de MVVM

El patrón de diseño MVVM es una arquitectura que separa el desarrollo de las aplicaciones móviles o web por capas, esto facilita la gestión del código y la escalabilidad de las aplicaciones, estos componentes son los siguientes:

- **Model (Modelo):** Esta capa tiene los datos de la aplicación, es decir que representa la lógica de negocio, contiene los métodos para la manipulación de datos tanto la importación como exportación de los mismos, para ello trabaja directamente con servicios trayendo los datos mediante los métodos API REST del Back-end.
- **View (Vista):** Esta capa es la interfaz con la que el usuario interactúa (UI), dentro de esta se presenta la información que el ViewModel transforma del Modelo, también captura la interacción del usuario para posteriormente enviarlas al ViewModel.
- **ViewModel (Modelo de Vista):** Esta capa actúa intermediario entre la Vista y el Modelo, dentro de esta capa se transforman los datos que se obtienen del modelo en información para que se puedan presentar posteriormente en la Vista, aquí se maneja la lógica de presentación.

4.2. Diseño de Base de datos

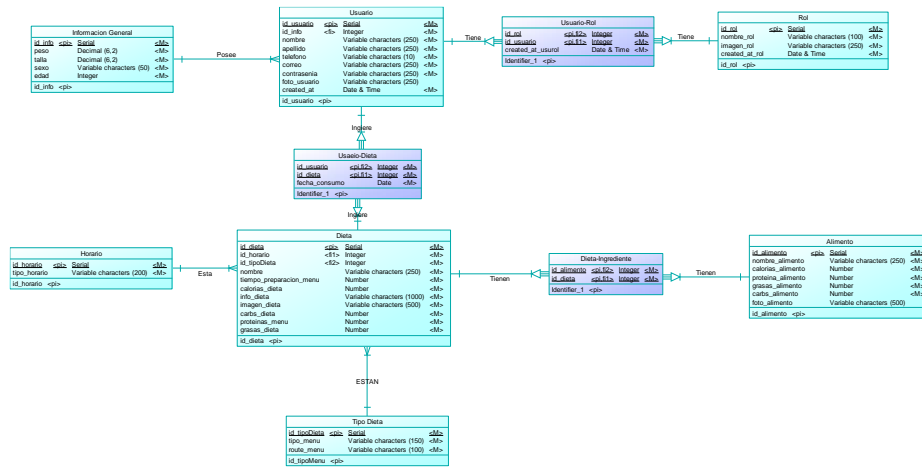
Para este punto se han considerado todos los campos necesarios así mismo al hacer una comparativa entre las bases de datos descritas en la tabla 3 y teniendo en cuenta que se trabajaran datos que se relacionan entre sí, se ha concluido que se utilizara una base de datos relacional, en este caso MySQL, que ayudará a gestionar todos los datos que se generarán dentro de la aplicación.

4.2.1. Diagrama Lógico

La Figura 10 muestra el diseño de la base de dato en donde después de crear las relaciones entre las tablas y establecer la cardinalidad de cada una, en este diagrama se obtiene nuevas tablas, cuando las cardinalidades son de muchos a muchos, al tener las tablas de Usuario, Información Nutricional, Rol, Dieta, Horario, Ingrediente, Tipo Menú, se obtuvieron 3 tablas nuevas en donde se tiene como Usuario-Rol que establece los roles que podría tener un usuario, de la misma manera al crear una Dieta se tienen un conjunto de alimentos que componen esa dieta Alimentos para lo cual se creó una tabla en donde se tengan los alimentos para cada Dieta, y para finalizar las dietas que se establecen al usuario una vez que se registra dentro de la aplicación, selecciona su objetivo y preferencias.

Figura 10

Diagrama Lógico de la Base de Datos



Autor. Elaborado por el Autor

5. Capítulo 5: Desarrollo

5.1. Estándares en el desarrollo de la aplicación

Se ha establecido estándares al momento del desarrollo de la aplicación tanto para la parte de Backend (Nest) como la parte del FrontEnd (Flutter), esto consta en la organización y el trabajo por módulos y capas, protegiendo así el paso de los datos como se muestra en la Figura 11 y en la Figura 12, utilizando así una buena práctica para el desarrollo de la parte del Back-End.

Figura 11

Servicio de la Clase Alimento

```
@Injectable()
export class AlimentosService {

  constructor(@InjectRepository(Alimento) private alimentoRepository:Repository<Alimento>)
  {}

  create(alimento: CreateAlimentoDDTO){
    const newAlimento = this.alimentoRepository.create(alimento);
    return this.alimentoRepository.save(newAlimento);
  }

  findAll(){
    return this.alimentoRepository.find();
  }

  async update(id:string,alimento:UpdateAlimentoDTO){

    const alimentoFound= await this.alimentoRepository.findOneBy({id_alimento:id});

    if(!alimentoFound){
      return new HttpException('Alimento No Existe',HttpStatus.NOT_FOUND);
    }

    const updatedUser=Object.assign(alimentoFound,alimento);

    return this.alimentoRepository.save(updatedUser);
  }
}
```

Autor. Elaborado por el Autor

Figura 12

Controlador de la Clase Alimento

```
@Controller('alimentos')
export class AlimentosController {

  constructor(private alimentoService: AlimentosService){}

  @Post()
  createAlimento(@Body() alimento: CreateAlimentoDDTO){

    return this.alimentoService.create(alimento);
  }

  @Get()
  findAll(){
    return this.alimentoService.findAll();
  }

  @Put('/:id')
  updateAlimento(@Param('id', ParseIntPipe) id, @Body() alimento: UpdateAlimentoDDTO){
    return this.alimentoService.update(id, alimento);
  }
}
```

Autor. Elaborado por el Autor

También se ha utilizado la encriptación de claves y datos que protegen la base de datos en el BackEnd, de la misma manera utilización de variables “utils” únicamente para la protección de la IP cuando se ejecuta tal como se observa en la Figura 13 y **¡Error! No se encuentra el origen de la referencia.**, esta práctica se la ha implementado ya que se quiere dar seguridad al BackEnd.

Figura 13

Método listen para el puerto y la IP

```
1 import { NestFactory } from '@nestjs/core';
2 import { AppModule } from './app.module';
3 import { ValidationPipe } from '@nestjs/common';
4 import { Ip_Address } from './utils/protectIp';
5
6 async function bootstrap() {
7   const app = await NestFactory.create(AppModule);
8   app.useGlobalPipes(new ValidationPipe());
9   await app.listen(3000, Ip_Address || "localhost");
10 }
11 bootstrap();
```

Autor. Elaborado por el Autor

Figura 14

Módulo Configuración de la Base de Datos

```
import { host_db, name_db, password_db, port_db, user_db } from './utils/protectDB';

@Module({
  imports: [
    TypeOrmModule.forRoot({
      type: 'mysql',
      host: host_db,
      port: port_db,
      username: user_db,
      password: password_db,
      database: name_db,
      entities: [Usuario,Rol,Alimento,Dieta,TipoMenu,Horario],
      synchronize: true,
    }),
    UsersModule,
    AuthModule,
    RolesModule,
    AlimentosModule,
    DietasModule,
    TipoMenuModule,
    HorarioModule,
  ],
})
```

Autor. Elaborado por el Autor

5.2.Organización de Módulos y Componentes

La organización tanto en el BackEnd como el FrontEnd fue desarrollada por módulos, siempre separando y trabajando de manera independiente cada una de las clases. De ser necesario, se exportaba la clase o clases al módulo que se estaba trabajando. De la misma manera para el FrontEnd se trabajó de manera separada, la capa de presentación (View), la parte de los estados y eventos (ViewModel) y la parte de la obtención de datos desde el API REST (Model).

Para la organización de componentes que se utilizó en la parte del FrontEnd, se cargaron los componentes gráficos(widgets) dentro de un archivo Stateless, para posteriormente cargarlos a un archivo contenedor que contienen estados a un archivo Stateful para capturar los cambios del UI. De la misma manera, se separaron en funciones dentro del archivo Stateless algunos componentes para tener un manejo más óptimo del código en la parte del FrontEnd.

5.3. Levantamiento de Requerimientos

5.3.1. Historias de Usuario

Para el levantamiento de requerimientos se hará el uso de historias usuarios para un manejo óptimo de cada uno de los módulos que se han tomado en consideración, así mismo poder analizar cada uno de las funcionalidades que tendrá cada módulo.

Nombre de Historias de Usuario	Módulo
Registro de Información Personal	Gestión de Registro de datos personales
Registro de Información Nutricional	
Edición de Información General del Usuario	
Login y Logout	Autenticación dentro de la aplicación
Presentación de dietas según los requerimientos	Gestión de dietas personalizadas
Administración de ingesta calórica diaria	
Administración de ingesta de agua diaria	Gestión de ingesta diaria de agua

Tabla 5. Tabla Historias de Usuario y Módulos

Autor: Realizada por el Autor

En base a la entrevista que se ha planteado se pudieron definir 6 historias de usuario las cuales son las partes fundamentales sobre las que tratará la aplicación que son las siguientes:

Historia de usuario N°1 – Registro de información del usuario:

Historia de Usuario	
Número: 1	Responsable: Usuario
Nombre de la historia: Registro de información del Usuario	
Prioridad del negocio: Alta	Riesgo en Negocio: Alta
Programador responsable: Erick Molina	
Descripción: El cliente puede registrarse dentro de la aplicación si no tiene una cuenta dentro de la misma, únicamente se le solicitarán los datos personales tales como nombre, apellido, teléfono, email, contraseña, edad, talla, género y peso.	
Observaciones: Está validado que el email tenga un formato correcto y la contraseña contiene una verificación, es decir un campo extra en donde se repita la contraseña para que el usuario se asegure de que digite bien la contraseña. Los datos se están almacenando en la Base de Datos de MySQL, tanto para la encriptación y desencriptación de la contraseña se utiliza la librería BCrypt que permite encriptar la contraseña al momento de enviarla a la BD, así mismo nos ayuda a desencriptarla cuando se la necesite.	

Tabla 6. Historia de Usuario Registro de información del usuario

Autor. Realizado por el Autor

Historia de usuario N°2 – Registro de información nutricional:

Historia de Usuario	
Número: 2	Responsable: Usuario
Nombre de la historia: Registro de Información Nutricional	
Prioridad del negocio: Alta	Riesgo en Negocio: Alta
Programador responsable: Erick Molina	
Descripción: Una vez el usuario haya registrado su información personal, el usuario registrara dentro de la aplicación su información nutricional, las cuales son su objetivo al que desea llegar, de la misma manera sus alimentos preferidos.	
Observaciones: El usuario elegirá unos de los 3 objetivos que se han planteado, los cuales son Ganar Masa Muscular, Perder Peso o Mantenerlo. Por otro lado, una vez elegido su objetivo, se registra los alimentos que el usuario prefiera, los usuarios tienen la opción de elegir entre las categorías de proteínas, carbohidratos, lácteos y frutas.	

Tabla 7. Historia de Usuario Registro de Información Nutricional

Autor: Realizado por el Autor

Historia de usuario N°3 – Edición de información general del usuario:

Historia de Usuario	
Número: 3	Responsable: Usuario
Nombre de la historia: Edición de Información General del Usuario	
Prioridad del negocio: Alta	Riesgo en Negocio: Alta
Programador responsable: Erick Molina	
Descripción: El usuario tiene la opción de poder editar su información general, una vez dentro de la pantalla de edición el usuario puede cambiar su foto de perfil, nombre, apellido, email, teléfono, talla, peso y edad.	
Observaciones: Para la edición de los datos del usuario se estableció un campo en donde se puede editar la foto del usuario, para ello se implementó un método que almacena la foto del usuario dentro de Firebase y devuelve el link, el cual es almacenado dentro de la base de datos de MySQL, por otro lado, se debe tomar en consideración que solamente el usuario puede editar su información, para lo cual se utilizará el token de sesión del usuario para poder hacerlo.	

Tabla 8. Historia de Usuario Edición de información general del usuario

Autor. Realizado por el Autor

Historia de usuario N°4 – Login y Logout:

Historia de Usuario	
Número: 4	Responsable: Usuario
Nombre de la historia: Login y Logout	
Prioridad del negocio: Alta	Riesgo en Negocio: Alta
Programador responsable: Erick Molina	
Descripción: El usuario puede hacer login y logout de acuerdo a únicamente si se registró previamente, para lo cual el ingreso debe ser con el correo y contraseña registrados.	
Observaciones: Se implementó validaciones tanto para la contraseña y para el email con el manejo de errores en caso de que no exista el email registrado o que la contraseña sea incorrecta, en caso de que el usuario no exista tiene la posibilidad de registrarse dentro de la aplicación.	

Tabla 9. Historia de Usuario Login y Logout

Autor: Realizado por el Autor

Historia de usuario N°5 – Presentación de dietas según los requerimientos:

Historia de Usuario	
Número: 5	Responsable: Sistema
Nombre de la historia: Presentación de dietas según los requerimientos	
Prioridad del negocio: Alta	Riesgo en Negocio: Alta
Programador responsable: Erick Molina	
Descripción: Después de que el usuario haya seleccionado su objetivo, se presentan una serie de dietas que tiene a su disposición para poder cumplir sus objetivos, cada una de las dietas que se presentan, están clasificadas por horario es decir desayuno, media comida, almuerzo y cena.	
Observaciones: Las dietas son generadas en base al objetivo elegido y los alimentos preferenciales del usuario, las dietas que se generarán tienen la información necesaria para que el usuario pueda tomar la mejor opción al momento de consumirlas, las cuales son: su forma de preparación, ingredientes tiempo y macronutrientes, en base a ellos el usuario puede tomar la opción que guste.	

Tabla 10. Historia de Usuario Presentación de dietas según los requerimientos

Autor. Realizado por el Autor

Historia de usuario N°6 – Administración de ingesta calórica diaria:

Historia de Usuario	
Número: 6	Responsable: Usuario/Sistema
Nombre de la historia: Administración de ingesta calórica diaria	
Prioridad del negocio: Medio	Riesgo en Negocio: Medio
Programador responsable: Erick Molina	
Descripción: Una vez que el usuario haya seleccionado la dieta según el horario se puede visualizar la cantidad de ingesta calórica diaria una vez que se haya registrado la dieta que se utilizó en la hora del día.	
Observaciones: El sistema puede mostrar al usuario cual es la cantidad de calorías mediante una barra de progreso, esto se dará siempre y cuando el usuario elija lo que va a ingerir en el horario que prefiera y las dietas que se han presentado en la Historia de Usuario N.º 5.	

Tabla 11. Historia de Usuario Administración de ingesta calórica diaria

Autor: Realizado por el Autor

Historia de usuario N°7 – Administración de ingesta de agua diaria:

Historia de Usuario	
Número: 7	Responsable: Usuario
Nombre de la historia: Administración de ingesta de agua diaria	
Prioridad del negocio: Baja	Riesgo en Negocio: Baja
Programador responsable: Erick Molina	
Descripción: El usuario puede registrar la ingesta de agua diaria que pueda tener, es únicamente un recurso opcional si el usuario lo desea.	
Observaciones: El usuario puede controlar el consumo de agua que obtiene en el día, se presenta un apartado en donde puede registrar su consumo y la aplicación almacena dicha información con el token de sesión creada y con el método de Flutter SharedPreferences que nos permite almacenar la información únicamente con el token de sesión y el id del usuario.	

Tabla 12. Historia de Usuario Administración de ingesta de agua diaria

Autor. Realizado por el Autor

5.4.Prototipos

5.4.1. Prototipo Gestión de Registro de Datos Personales

Para desarrollar el prototipo de gestión sobre el registro de datos personales se contó con las historias de usuario, las cuales son: registro de información del usuario, registro de información nutricional y la edición de la información general, todas ellas registradas en el módulo de gestión de registro de datos personales.

5.4.1.1. Desarrollo del Prototipo

Al desarrollar la aplicación se tomó en cuenta los parámetros detallados en las historias de usuario, las cuales fueron el registro dentro de la aplicación y la edición, estas al formar parte del módulo de registro de datos personales fueron desarrolladas de acuerdo a los requerimientos solicitados.

Para el desarrollo del Back-end se utilizó como herramienta NestJs, que es framework del lenguaje de programación de JS que lo que permitió crear la entidad del usuario en donde contenía toda la información que se requiere para el registro, de la misma manera se creó funciones en la parte de los servicios los cuales fueron put y post.

Una vez creado los métodos del servicio las funciones fueron utilizados en la capa del controlador, para poder cuidar el paso de datos, los métodos que se crearon en el controlador fueron utilizados en la parte del Front-end que es Flutter, utilizando los métodos en las ventanas pertenecientes al registro del usuario.

5.4.1.2.Pruebas del Prototipo

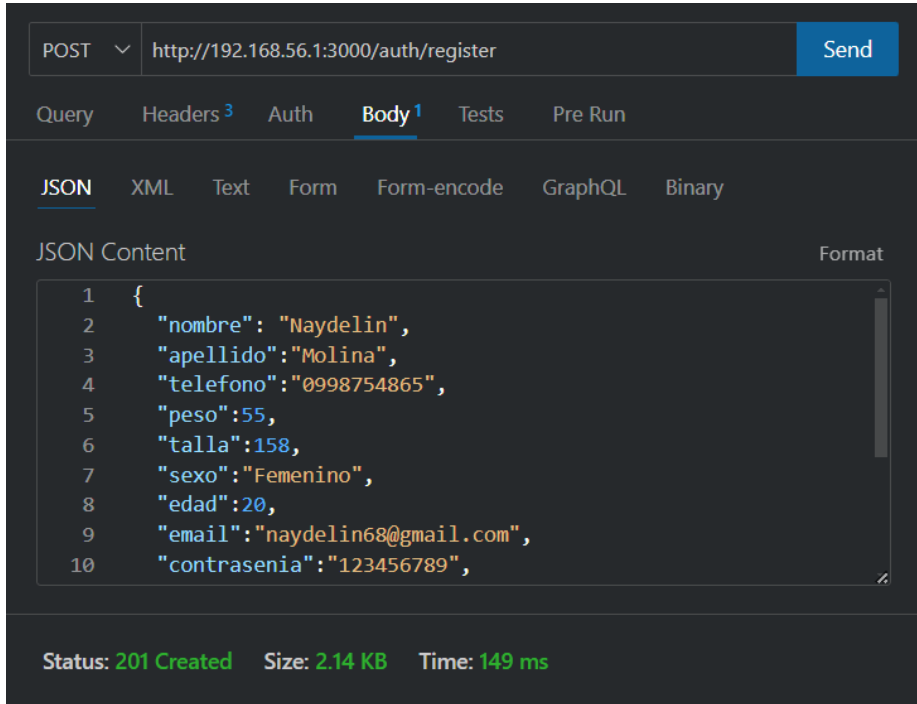
Las pruebas que se desarrollaron dentro de este módulo fueron las pruebas de integración y las pruebas de carga, como se detalló previamente en los criterios de evaluación. Las pruebas se hicieron dentro del Back-end a los métodos API REST de este módulo.

Pruebas de Integración:

La **Figura 15** representa la prueba de integración realizada al método POST al registro del usuario, cómo se puede observar dio un status correcto, esto quiere decir que el método post funciona correctamente para la parte de registro dentro de la aplicación.

Figura 15

Método Post del Módulo de Registro

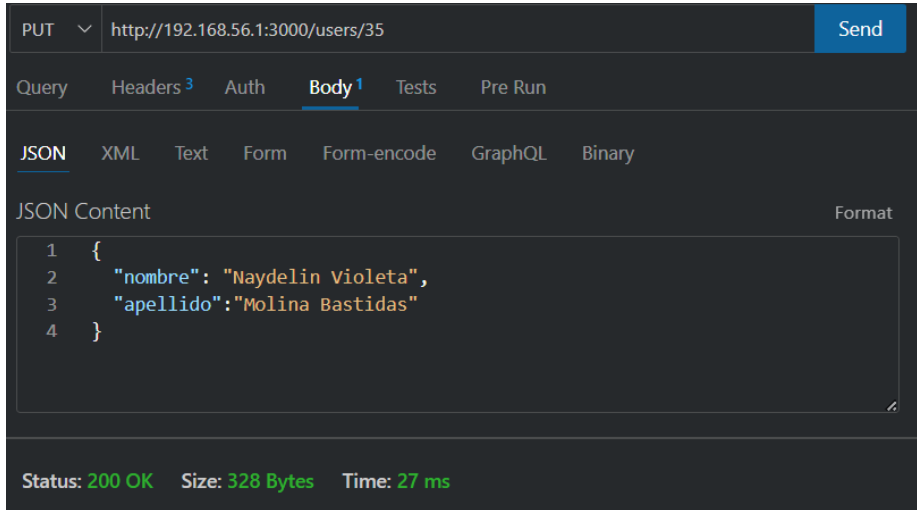


Autor. Elaborado por el Autor

La Figura 16 representa la prueba de integración desarrollada para el método PUT, esta se realizó para probar el funcionamiento de la edición de los datos registrados, como se puede observar tiene un status 200, esto significa que la edición de los datos fue correcta.

Figura 16

Método Put para el usuario



Autor. Elaborado por el Autor

Los resultados esperados al realizar las pruebas de integración fueron las que se deseaban, ya que todos los métodos del API REST trabajan de manera correcta, se pueden apreciar en cada una de las figuras como su status es de 200 o 201, es decir que la petición se efectuó de la manera correcta, ya sea que se creó el registro o que se modificó los datos del usuario.

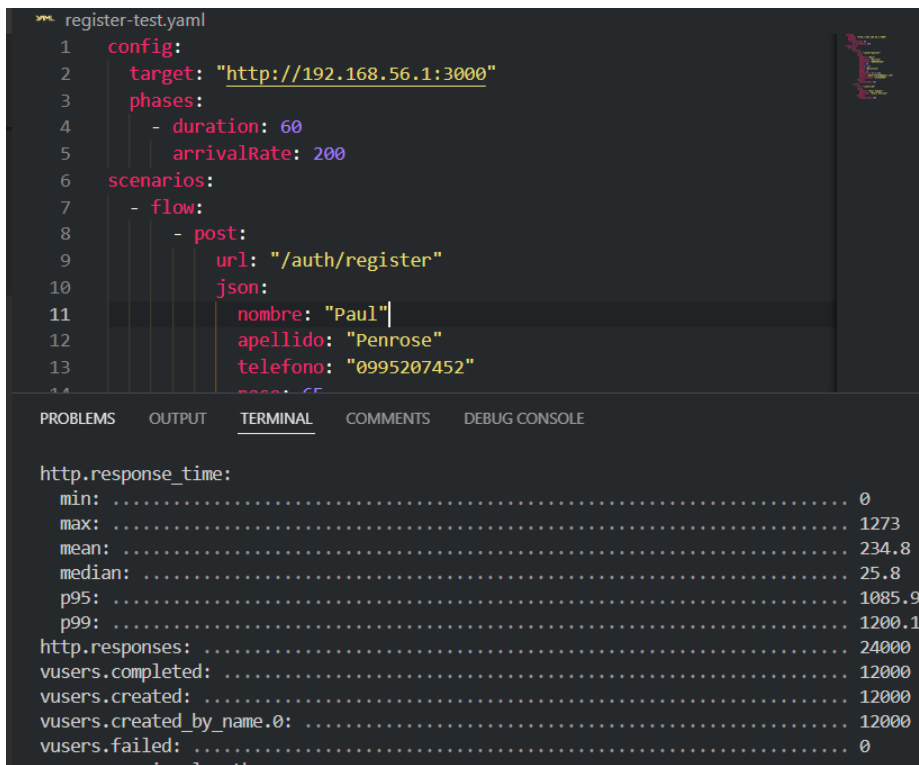
Pruebas de Carga:

La Figura 17 representa la prueba de carga que se realizó a los métodos API que fueron POST y PUT, para ello se consideró tomar un tiempo de prueba de 60

segundos y un total de 200 usuarios a la vez, las pruebas se realizaron utilizando los segundos por usuarios dando un total de 1200 usuarios ficticios probando a la vez. Observando la figura en el tiempo y usuarios establecidos no presentó un problema para el sistema, pero si se considera subir la carga de usuarios en el mismo tiempo fallará.

Figura 17

Prueba de Carga al Módulo de Registro



```
register-test.yaml
1  config:
2    target: "http://192.168.56.1:3000"
3    phases:
4      - duration: 60
5        arrivalRate: 200
6  scenarios:
7    - flow:
8      - post:
9        url: "/auth/register"
10       json:
11         nombre: "Paul"
12         apellido: "Penrose"
13         telefono: "0995207452"
14         correo: "penrose@gmail.com"

PROBLEMS  OUTPUT  TERMINAL  COMMENTS  DEBUG CONSOLE

http.response_time:
min: ..... 0
max: ..... 1273
mean: ..... 234.8
median: ..... 25.8
p95: ..... 1085.9
p99: ..... 1200.1
http.responses: ..... 24000
vusers.completed: ..... 12000
vusers.created: ..... 12000
vusers.created_by_name.0: ..... 12000
vusers.failed: ..... 0
vusers.session_timeout: ..... 0
```

Autor. Elaborado por el Autor

5.4.1.3.Feedback del Usuario

En base a la información aportada por el usuario se recolecto el feedback, el cual fue que dentro de la pantalla de registro de datos no se deberá digitar manualmente por el usuario la talla, el peso y la edad, ya que se podría presentar fallas al momento de digitar o entradas erróneas de datos que no puedan aportar al momento del cálculo de los macronutrientes que se deben ingerir.

5.4.1.4.Refinación del Prototipo

De acuerdo a la retroalimentación dada por el usuario, en este punto se consideró el cambio y mejora del diseño de entrada de los datos, para ello se utilizó un DatePicker propio de Flutter para la precisión en la entrada de datos.

5.4.2. Prototipo Autenticación dentro de la Aplicación

Para el desarrollo de este prototipo se contó con la información de las historias de usuarios realizadas, las cuales fueron Login y Logout de la aplicación consideradas dentro del módulo de autenticación.

5.4.2.1.Desarrollo del Prototipo

Para el desarrollo del prototipo se consideró los requerimientos del usuario, los cuales fueron el ingreso mediante correo y clave para la aplicación, también para la parte del logout del sistema se utilizó un icono ubicado en la pantalla del “Perfil” del usuario para que pueda ser mucho más intuitivo y fácil al momento de salir de la aplicación.

En este módulo, se desarrolló la parte del Back-end como un método post, lo cual permitía enviar una solicitud a la base de datos en donde devolvía toda la información del usuario si se encontraba registrado, para este método también se implementó un token de sesión, para que al momento que haga login dentro de la aplicación pueda tener un inicio seguro dentro de la aplicación, de la misma manera cada uno de estos tokens tendrá una duración de 24 horas, después de ese tiempo el token expirará, esto permita el uso seguro de cada una de las peticiones que se hagan en la aplicación.

Por otro lado, para el logout del usuario se manejó un método en el Front-end, la cual después de obtener al usuario logueado se presionará el botón de salir solamente si el usuario lo desea, cuando suceda eso se eliminará la sesión del usuario que se almacenó previamente con un método llamado sharedPreferences con una función remove, para que pueda quitar al usuario de la sesión y lo envíe de vuelta a la página de Login.

5.4.2.2. Pruebas del Prototipo

Para las pruebas de este prototipo se basó en el método POST creado para la parte del login, para ello se utilizó las mismas herramientas utilizadas en el prototipo anterior que fueron ThunderClient y Artillery para las pruebas de carga.

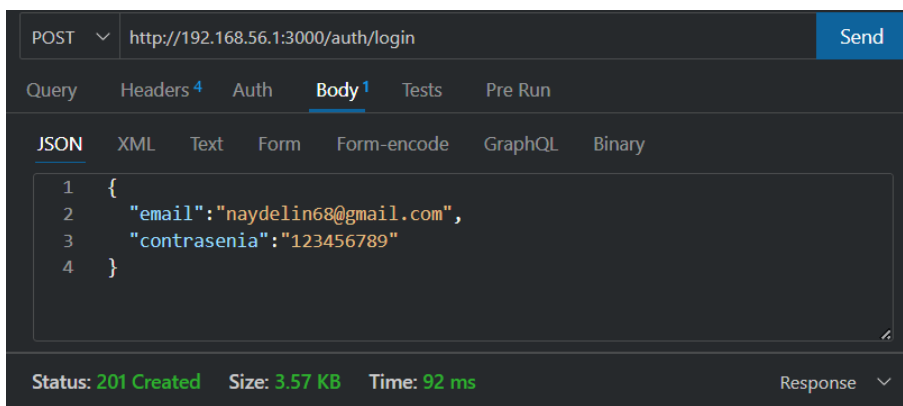
Pruebas de Integración:

Se puede observar dentro de la **Figura 18** el manejo del método POST dentro del login, el status para esta prueba fue exitoso, por ende, este método también

funciona a la perfección y no tiene ninguna interferencia o error con la parte de llamada de datos del login.

Figura 18

Método Post de Login del Sistema



Autor. Elaborado por el Autor

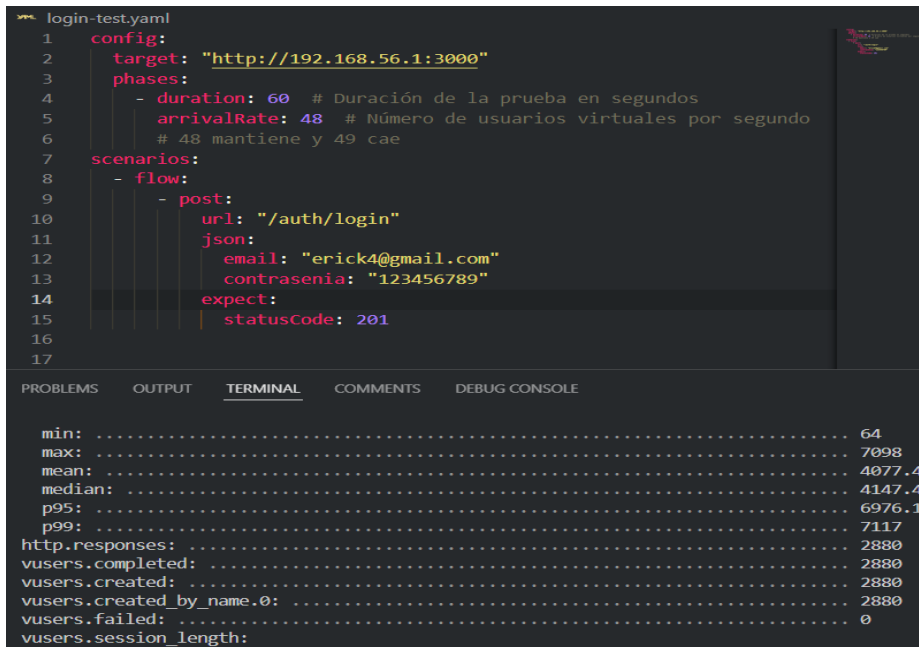
La prueba de integración realizada a este módulo fue exitosa, debido a que se utilizaron las credenciales de un usuario que estaba registrado previamente dentro de la aplicación, esta prueba al ser un método post, se puede apreciar que el status es de 201, lo que significa que la petición se realizó correctamente dentro del back-end, por lo que, al momento de llamar este método con las credenciales, se pudo obtener todos los datos del usuario que está registrado.

Pruebas de Carga:

La **Figura 19** representa la prueba de carga realizada, para ello se consideró un tiempo de 60 segundos y de 48 usuarios por segundo que fue la carga máxima que pudo aguantar, dando así un estimado de 2880 usuarios que fueron permitidos antes de que la aplicación fallará, pues después de ese número de usuarios en el tiempo de 60 segundos la aplicación empezará a presentar problemas y errores a los usuarios que estén manejando la aplicación.

Figura 19

Prueba de carga al Módulo Login



```
login-test.yaml
1  config:
2  target: "http://192.168.56.1:3000"
3  phases:
4  - duration: 60 # Duración de la prueba en segundos
5    arrivalRate: 48 # Número de usuarios virtuales por segundo
6    # 48 mantiene y 49 cae
7  escenarios:
8  - flow:
9    - post:
10     url: "/auth/login"
11     json:
12       email: "erick4@gmail.com"
13       contrasenia: "123456789"
14     expect:
15       statusCode: 201
16
17

PROBLEMS  OUTPUT  TERMINAL  COMMENTS  DEBUG CONSOLE

min: ..... 64
max: ..... 7098
mean: ..... 4077.4
median: ..... 4147.4
p95: ..... 6976.1
p99: ..... 7117
http.responses: ..... 2880
vusers.completed: ..... 2880
vusers.created: ..... 2880
vusers.created_by_name.0: ..... 2880
vusers.failed: ..... 0
vusers.session_length:
```

Autor. Elaborado por el Autor

5.4.2.3.Feedback del Usuario

Para el prototipo de este módulo no hubo un feedback presentado por parte del usuario, ya que el prototipo cumplió con las expectativas del usuario, y su experiencia con el inicio y cierre de sesión de la aplicación fue satisfactoria.

5.4.2.4.Refinación de Prototipo

Al no tener una retroalimentación por parte del usuario se continuó con el desarrollo del siguiente módulo, no hubo ningun cambio dentro de la aplicación ni en la parte del Back-end, ni del Front-end, todo esto debido a que el prototipo cumplió con los requerimientos que el usuario solicitaba y su experiencia fue satisfactoria en el módulo trabajado.

5.4.3. Prototipo Gestión de Dietas Personalizadas

Para el desarrollo de este módulo se trabajó con las historias de usuario como se ha trabajado los prototipos anteriores, dentro de este módulo se manejó la presentación de dietas según los requerimientos del usuario, los cuales se basaron en el objetivo y en las preferencias alimenticias, también la administración de ingesta calórica que tendrá que ver con las dietas que se le generen al usuario.

5.4.3.1.Desarrollo del Prototipo

Para el desarrollo del prototipo se trabajó con el módulo de gestión de datos personales que se desarrolló en la primera iteración, estos datos fueron considerados importantes debido a que se necesitó los datos del usuario para generar dietas, los cuales fueron peso, talla, edad, genero, objetivo y preferencias alimenticias, todo esto

con el fin de que se puedan generar dietas satisfactoriamente de acuerdo a los requisitos que el usuario presente.

Se tomó en cuenta para la generación de dietas los métodos de filtrado propios de JS aplicándolas en el Front-end, en el cual se pasaron los alimentos que fueron elegidos por el usuario y las dietas según el objetivo, en donde se filtró las dietas mediante los alimentos con el método filter, también se debe considerar que el filtrado fue posible ya que la entidad de dietas tiene como foreign key la tabla de alimentos y el objeto JSON al llamar a la tabla dietas presenta los datos de la dieta junto a los alimentos que la conforman.

Finalmente, una vez obtenidas las dietas, se proceden a mostrar en el Front-end mediante cards, de la misma manera, se clasifican por horario, esto también fue posible ya que la tabla de dietas tiene el id del horario establecido en cada una de las dietas.

5.4.3.2. Pruebas del Prototipo

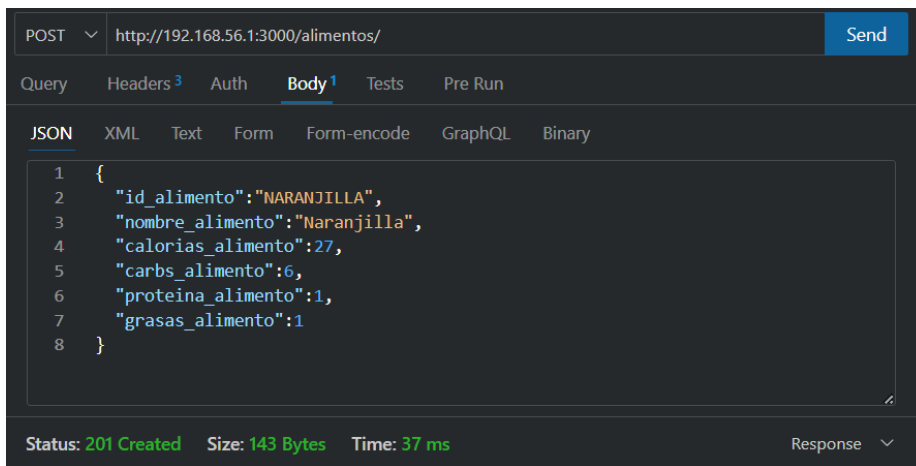
Para las pruebas de este prototipo se utilizaron los métodos de POST y GET de los alimentos y para la parte de dietas se manejó el método GET con el parámetro de objetivo para la prueba de carga y también se consideró el método pero también agregando el método POST para la prueba de integración, todo esto con el fin de ver si los métodos trabajan de forma correcta tanto en las pruebas de integración como en las pruebas de carga, estos métodos fueron importantes ya que se trabajaron dentro de la parte del Front-end para obtener toda la información tanto de los alimentos para que el usuario los pueda elegir los de su preferencia como los datos de las dietas que se le presentarán al usuario.

Pruebas de Integración:

La Figura 20 representa la llamada del método POST de alimentos, en donde se puede observar que el método funciona correctamente y que no tiene fallo en el método al momento de crear un nuevo alimento.

Figura 20

Método Post de Alimentos



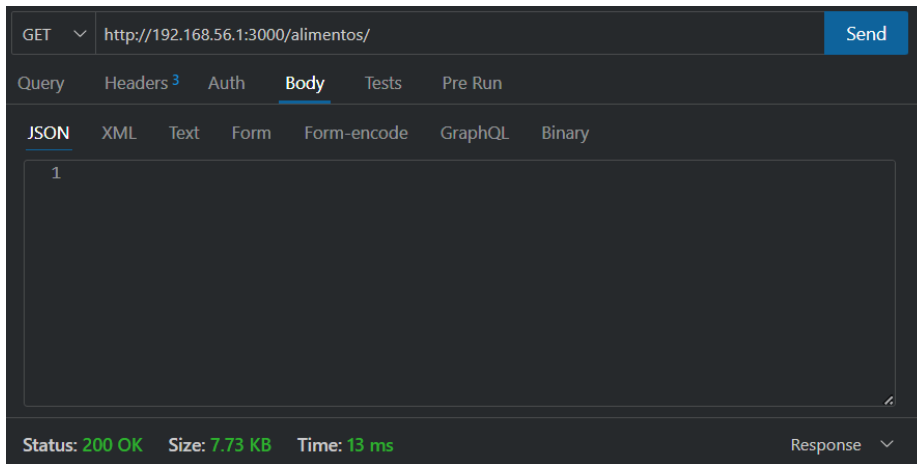
Autor. Elaborado por el Autor

En la

Figura 21 se representa el método GET que funciona de una manera correcta al momento de llamar a todos los alimentos que se han almacenado dentro de la base de datos, esto nos servirá para mostrar al usuario los alimentos y que pueda elegir los que el usuario prefiera.

Figura 21

Método Get de Alimentos



Autor. Elaborado por el Autor

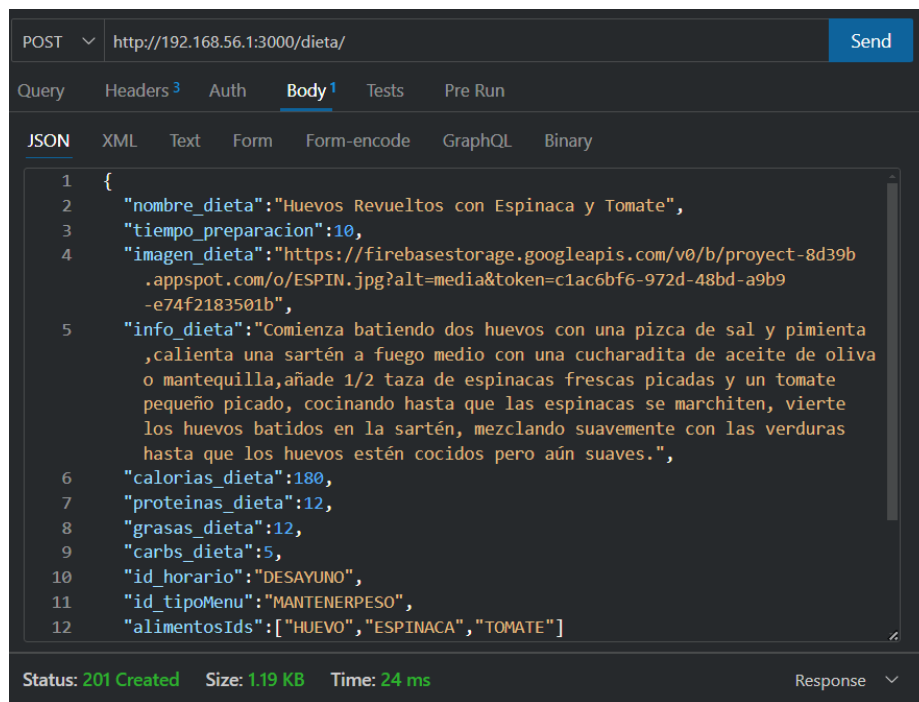
En la

Figura 22 representa la prueba que se realizó con el método POST para la parte del ingreso de dietas, en donde se puede observar que el método funciona de manera correcta y que se también maneja los ids de los alimentos que contiene, esto hace que

al momento de elegir los alimentos preferidos se filtren por los ids de cada dieta y pueda generar al usuario las dietas que tengan sus preferencias.

Figura 22

Método Post de Dietas



```
POST http://192.168.56.1:3000/dieta/ Send
Query Headers 3 Auth Body 1 Tests Pre Run
JSON XML Text Form Form-encode GraphQL Binary
1 {
2   "nombre_dieta": "Huevos Revueltos con Espinaca y Tomate",
3   "tiempo_preparacion": 10,
4   "imagen_dieta": "https://firebasestorage.googleapis.com/v0/b/proyect-8d39b
5     .appspot.com/o/ESPIN.jpg?alt=media&token=c1ac6bf6-972d-48bd-a9b9
6     -e74f2183501b",
7   "info_dieta": "Comienza batiendo dos huevos con una pizca de sal y pimienta
8     ,calienta una sartén a fuego medio con una cucharadita de aceite de oliva
9     o mantequilla,añade 1/2 taza de espinacas frescas picadas y un tomate
10    pequeño picado, cocinando hasta que las espinacas se marchiten, vierte
11    los huevos batidos en la sartén, mezclando suavemente con las verduras
12    hasta que los huevos estén cocidos pero aún suaves.",
13   "calorias_dieta": 180,
14   "proteinas_dieta": 12,
15   "grasas_dieta": 12,
16   "carbs_dieta": 5,
17   "id_horario": "DESAYUNO",
18   "id_tipoMenu": "MANTENERPESO",
19   "alimentosIds": ["HUEVO", "ESPINACA", "TOMATE"]
20 }
```

Status: 201 Created Size: 1.19 KB Time: 24 ms Response

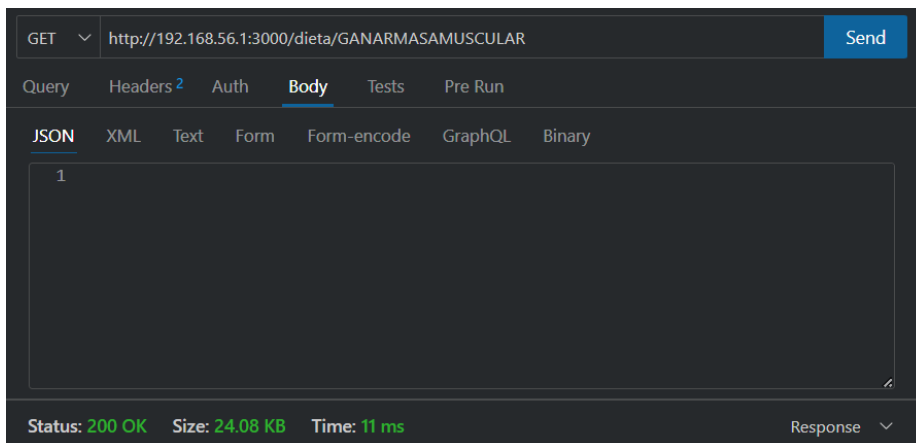
Autor. Elaborado por el Autor

Para esta prueba se utilizó el método GET que se representa en la

Figura 23, pero pasando un parámetro, este parámetro será el id del objetivo que el usuario elige, este método funciona de manera correcta, ya que es muy importante para obtener todas las dietas de acuerdo ese id para posteriormente filtrarlas por los alimentos que se elijan.

Figura 23

Método Get de Dietas según el objetivo



Autor: Elaborado por el Autor

Las pruebas de integración que se realizaron los métodos API REST de este módulo fueron ejecutadas correctamente y no presentaron ningún tipo de error, ya que como se pudieron observar en cada una de las figuras presentadas presentan status de

200 y 201, lo que nos dice que todas las peticiones fueron realizadas correctamente y sin ningún tipo de problema al momento de cargar y obtener los datos desde la BD.

Pruebas de Carga:

La

Figura 24 representa la prueba de carga que se realizó a los métodos POST y GET de la parte de alimentos para ello se utilizó el mismo tiempo que las anteriores pruebas que fue de 60 segundos, pero los usuarios que soporta antes de que los métodos fallen es de 180 usuarios por segundo, dando así un total de 10800 usuarios en donde la aplicación permite usar estos métodos de manera simultánea.

Figura 24

Prueba de Carga de Alimentos

```
load-test.yaml
1  config:
2    target: "http://192.168.56.1:3000"
3    phases:
4      - duration: 60
5        arrivalRate: 180
6    scenarios:
7      - flow:
8        - post:
9          url: "/alimentos"
10         json:
11           id_alimento: "YOGURTGRIEGO"
12           nombre_alimento: "Yogurt Griego"
13           calorias_alimento: 59
14           proteina_alimento: 17
15           carbs_alimento: 4
16           grasas_alimento: 1
17         expect:
18           statusCode: 201

PROBLEMS  OUTPUT  TERMINAL  COMMENTS  DEBUG CONSOLE

mean: ..... 2384.1
median: ..... 1901.1
p95: ..... 6439.7
p99: ..... 6976.1
http.responses: ..... 32400
vusers.completed: ..... 10800
vusers.created: ..... 10800
vusers.created_by_name.0: ..... 10800
vusers.failed: ..... 0
```

Autor. Elaborado por el Autor

Se representa en la Figura 25 la prueba de carga realizada a las dietas, para ello se consideró los métodos GET, GET con el parámetro del objetivo y POST, al realizar las pruebas también se consideró el tiempo de 60 segundos y un total de 200 usuarios que permitía el sistema dentro del tiempo que se ha establecido, presentando un total de 12000 mil usuarios utilizando los métodos en el tiempo sin que presenten fallos.

Figura 25

Prueba de Carga de Dietas

```
diatas-test.yaml
1  config:
2    target: "http://192.168.56.1:3000"
3  phases:
4    - duration: 60
5      arrivalRate: 200
6  escenarios:
7    - flow:
8      - get:
9          url: "/dieta"
10         expect:
11           statusCode: 200
12       - get:
13         url: "/dieta/PERDERPESO"
14         expect:
15           statusCode: 200
16       - post:
17         url: "/dieta"
18         expect:
19           statusCode: 201

PROBLEMS  OUTPUT  TERMINAL  COMMENTS  DEBUG CONSOLE

http.request_rate: ..... 200/sec
http.requests: ..... 12000
http.response_time:
  min: ..... 2
  max: ..... 4513
  mean: ..... 1835.2
  median: ..... 1587.9
  p95: ..... 3905.8
  p99: ..... 4492.8
http.responses: ..... 12000
vusers.completed: ..... 12000
vusers.created: ..... 12000
vusers.created_by_name.0: ..... 12000
vusers.failed: ..... 0
```

Autor. Elaborado por el Autor

5.4.3.3. Feedback del Usuario

El usuario presento el feedback para este módulo, el cual sugirió mejorar el diseño para la parte del contador de calorías y también poniendo la informacion de

macronutrientes bajo la foto de la dieta para saber cuál será su mejor opción para consumir de acuerdo a los macronutrientes que se presentan en las dietas.

5.4.3.4. Refinación del Prototipo

Se realizó la refinación del prototipo de acuerdo al feedback que el usuario presentó, una vez realizado el refinamiento se procedió a mostrar nuevamente el prototipo al usuario, en donde lo aprobó y dando así el paso al desarrollo del siguiente prototipo.

5.4.4. Prototipo Gestión de Ingesta de Agua

Para este prototipo se desarrolló el apartado de ingesta de agua, para que el usuario pueda gestionar su consumo diario, para lo cual se realizará un contador de vasos ingeridos al día, esto se almacenará de manera local dentro de la aplicación, para este prototipo no se realizaran pruebas de carga o de integración, ya que no se posee métodos dentro del Back-end.

5.4.4.1. Desarrollo del Prototipo

Este prototipo se desarrolló en base a los requerimientos que se recolectaron, para cual se desarrolló un apartado dentro de la ingesta de dietas y el contador de calorías en donde se agregó un widget que el usuario puede manipular para registrar el consumo de agua diaria.

Para el desarrollo de este módulo, se trabajó directamente en el Front-end, para lo que se utilizó `sharedPreferences` que es un método propio de Flutter, lo que nos permitió es poder almacenar la información únicamente de las calorías del usuario y los

vasos de agua que consumió el usuario, de esta manera, en caso de que el usuario cierre la sesión se almacene la información de la ingesta de calorías y de agua.

De esta manera, una vez que el usuario quiera volver a ingresar a la aplicación, los datos que fueron almacenados se verán reflejados en la aplicación, los datos se almacenarán únicamente con el id del usuario, así mismo, al completar la ingesta de macronutrientes diarios las barras de progreso se reiniciarán para que el usuario pueda volver a completar su ingesta de macronutrientes al siguiente día, de la misma manera sucederá con el contador de vasos de agua.

5.4.4.2.Feedback del Usuario

El usuario proporcionó su feedback en donde manifestó que se debe explicar al usuario la cantidad de vasos diarios que debe tomar, así como la cantidad de agua que debe ingerir en cada vaso, también se debe tener en cuenta que el widget debe ser muy intuitivo para que el usuario lo pueda manejar de manera correcta.

5.4.4.3.Refinamiento del Prototipo

El refinamiento del prototipo se reconstruyó el diseño para el contador de agua, para ello se usó iconos que hacían alusión a un vaso de agua en donde el usuario puede presionar y se registra su consumo de agua, también lo puede quitar si lo desea, además se agregó un cuadro de información en donde se detalla al usuario la cantidad de agua que debe contener cada vaso que ingiera.

5.5.Muestra de la Aplicación

5.5.1. Registro dentro de la Aplicación

Dentro del Registro de la aplicación se tienen los datos más importantes que se consideraron importantes que se utilizaron dentro del sistema, los datos del usuario y su información biométrica los cuales se utilizan para calcular cuantos macronutrientes el usuario debe comer en el día. De la misma manera se tienen los datos que van a ser esenciales para el acceso a la aplicación como son el correo y contraseña como se muestra en la Figura 26.

Figura 26

Pantalla de Registro de la Aplicación

←

Regístrate

Nombre
Naydelin

Apellido
Bastidas

Telefono
0998541257

Email
nayde@hotmail.com

Contraseña
.....

Confirmar Contraseña
.....

Registrar

←

Información General

Edad 25 años

Talla 1.65 cm

Peso 60 kg

Sexo ♂ Masculino

Ingresar

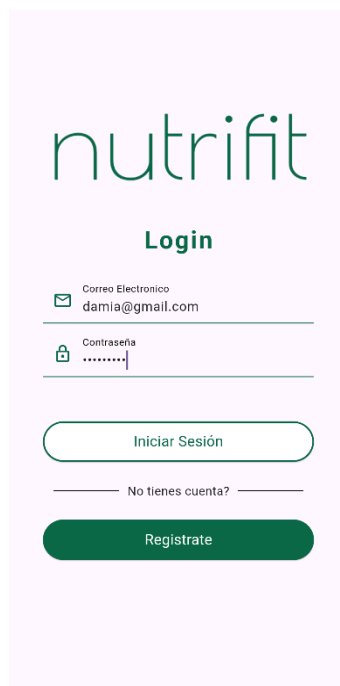
Autor. Elaborado por el Autor

5.5.2. Login de la Aplicación

En la Figura 27 se muestra la pantalla del login de la aplicación una vez que el usuario se haya registrado podrá acceder con las credenciales que se almacenaron, también se debe tener en cuenta que si el usuario quiere acceder sin tener una cuenta se mostrará un mensaje de advertencia al no existir un correo o en caso de que la clave sea incorrecta, también se recomendará registrarse en caso de no estar registrado.

Figura 27

Pantalla de Login de la Aplicación



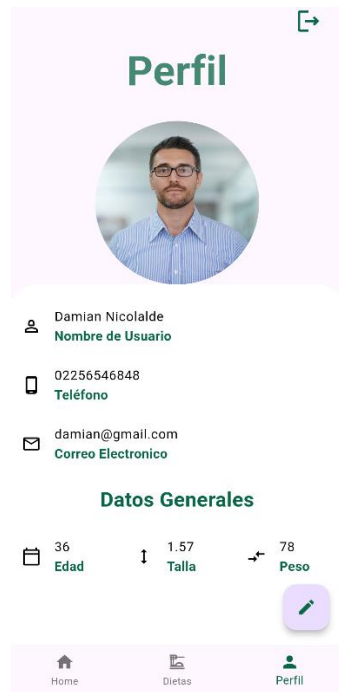
Autor. Elaborado por el Autor

5.5.3. Actualización del Perfil de Usuario

La Figura 28 representa la pantalla de perfil, donde se tienen los datos del usuario, también tendrá la posibilidad de poder editar su información si lo desea, de la misma manera se podrá cargar una foto del usuario en caso de que quiera hacerlo, también el usuario podrá hacer logout si lo desea.

Figura 28

Pantalla de Actualización de Perfil de la Aplicación



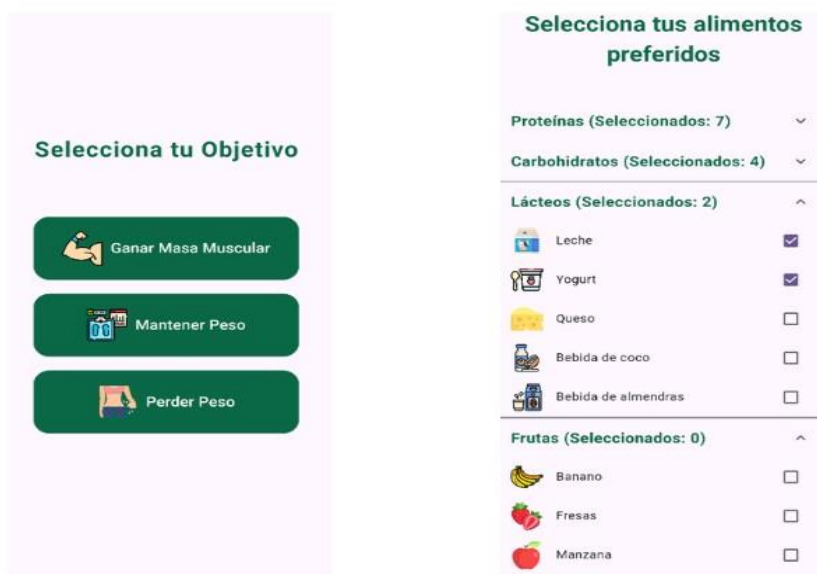
Autor. Elaborado por el Autor

5.5.4. Elección de Objetivo y Preferencias

En la Figura 29 se muestran las pantallas de elección de objetivo el usuario podrá elegir una de las 3 opciones que se presentan, posteriormente se presentara otra pantalla en donde se tienen una lista de alimentos que el usuario podrá elegir entre las que se sean de su preferencia, para ello el usuario necesariamente deberá elegir 3 de cada uno para que pueda tener una variedad al momento de generar dietas.

Figura 29

Pantalla de Elección de Objetivos y Preferencias de la Aplicación



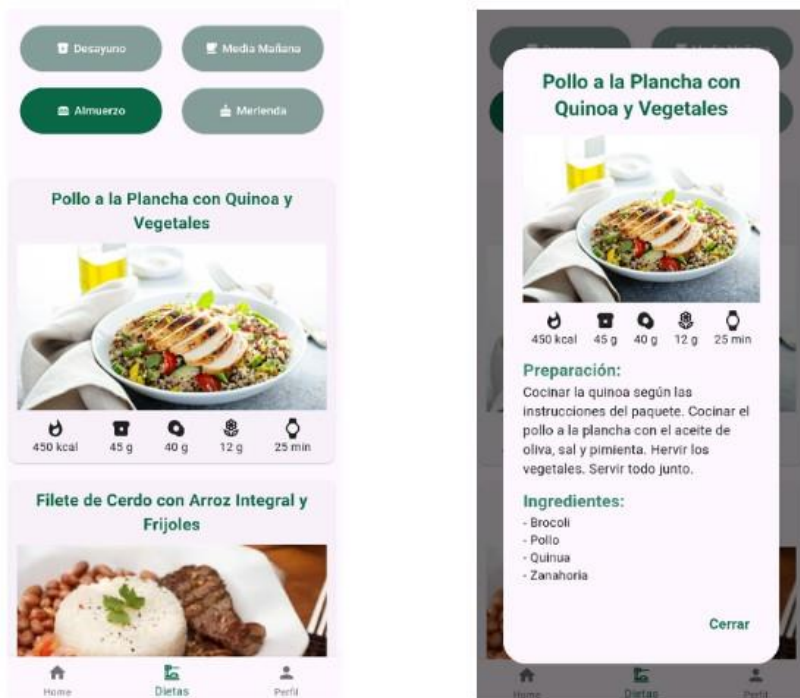
Autor. Elaborado por el Autor

5.5.5. Generación de Dietas Personalizadas

Una vez que se hayan elegido los objetivos y los alimentos de preferencia, se generaran dietas de acuerdo a sus elecciones, las cuales se presentaran por horario de comida, así mismo se tendrá información sobre el tiempo de preparación, calorías que contiene, proteínas, carbohidratos y grasas como se puede apreciar en la **¡Error! No se encuentra el origen de la referencia..** Si se desea conocer más información, se deberá hacer tap en la dieta deseada, en donde se mostrará más información, como la forma de preparar y los ingredientes que contiene.

Figura 30

Pantalla de Generación de Dietas de la Aplicación



Autor. Elaborado por el Autor

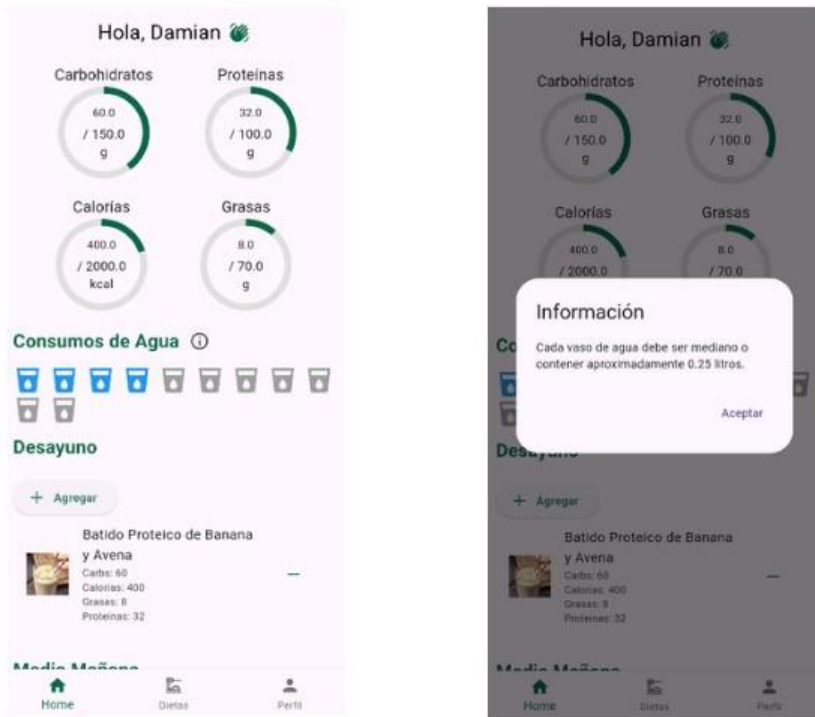
5.5.6. Consumos de Dietas, Contador de Macronutrientes y Contador de Agua

Se presentará una página principal en donde se tendrá los macronutrientes que se han consumido durante el día, estas se calculan de acuerdo a lo que el usuario ingiera de acuerdo a su horario, por otro lado, se tiene también el consumo de agua del usuario, como un opcional

se agregó la información sobre la cantidad que se debe ingerir para que el usuario lo tome en consideración, tal como se muestra en la Figura 31.

Figura 31

Pantalla de Consumo de Dietas y Contadores de la Aplicación



Autor. Elaborado por el Autor

6. Conclusiones y Recomendaciones

6.1. Conclusiones:

- Por medio de las reuniones y entrevistas que se tuvo con la persona del área de nutrición, se pudo recopilar toda la información necesaria para el desarrollo de la aplicación y se obtuvo todos los requerimientos necesarios para poderlos implementar. De la misma manera se facilitó información nutricional que aporta valor a la aplicación como lo son: el cálculo para consumo de macronutrientes de acuerdo a sus información general y objetivo, también el consumo de agua diario y también que alimentos se debe considerar para poder generar dietas personalizadas a cada uno de los usuarios.
- Gracias al desarrollo de esta aplicación se pueden generar dietas personalizadas en base a los parámetros solicitados al usuario en el momento de registrarse dentro de la aplicación, como lo son peso, altura, edad, sexo, preferencias alimenticias y el objetivo que se elige. Esto se ha logrado conjuntamente con la ayuda de la persona del área de nutrición, que ha proporcionado la información suficiente para llevar a cabo y poder aplicar todo el conocimiento obtenido dentro de la aplicación para que pueda cumplir con el objetivo de generar dietas.
- El desarrollo del prototipo de la aplicación móvil al ser desarrollada con el Framework Flutter de Dart permite que la aplicación pueda ser utilizada tanto en dispositivos Android y IOS, esto permite la accesibilidad a todos los usuarios. Por lo tanto, el uso de este Framework fue muy beneficioso en el desarrollo de esta aplicación móvil, ya

que no hubo necesidad de crear 2 proyectos distintos para cada uno de estos sistemas operativos.

- El uso de la metodología de Prototipado Evolutivo, junto con las tecnologías como lo fueron Flutter para el desarrollo móvil híbrido y NestJS para el manejo del back-end, fueron esenciales para desarrollar de manera exitosa la aplicación de gestión de nutrición y dietas. Por otra parte, la buena integración de los datos ha sido fundamental para poder generar las dietas según los objetivos y preferencias alimenticias de una manera eficiente, de esta manera, asegurando que la aplicación cumpla con los requerimientos y expectativas establecidas.
- El desarrollo de los prototipos incluyó la fase de evaluación y feedback del usuario, en donde se probó cada iteración y se pudo obtener la retroalimentación respectiva de cada módulo desarrollado, esto fue muy importante ya que se pudo corregir errores e implementar las mejoras que se han sugerido. Por otra parte, al realizar la fase de pruebas se pudo comprobar que la integración de cada uno de los métodos que se utilizaron fue correcta, de la misma manera, que la exportación e importación de datos fue exitosa.

6.2.Recomendaciones:

- El desarrollo de aplicaciones móviles es uno de los nichos de mercado que más se explora en estas épocas, por lo cual las herramientas que se elijan deben ser muy importantes y se debe tener en cuenta el tiempo que se desea emplear y los costes que se tiene para desarrollar, se si desea ahorrar estas características se debe emplear herramientas de desarrollo híbridas ya que al desarrollar de manera de nativa se debería tener en cuenta que el desarrollo es para cada uno de los sistemas operativos para los cuales se contempla implementar, para esto se debe tener en cuenta que si se desea desarrollar para Android y IOS se deberían desarrollar 2 proyectos independientes para cada uno de estas plataformas, aumentado así el costo y el tiempo de desarrollo. Por lo tanto, se debería tener en cuenta que tipo de aplicación se va a desarrollar y sobre todo revisar que características va a poseer y si se va a desarrollar de manera nativa o híbrida.
- La aplicación móvil se ha desarrollado en base a los requerimientos levantados, sin embargo, es una aplicación a la que se pueden implementar más módulos, ya que el tema de nutrición es basto y se podrían aumentar más características que puedan complementar a este proyecto. Se recomienda a futuros desarrollos de aplicaciones móviles poder ampliar todo el tema de desarrollo de aplicaciones con el tema de nutrición, con la finalidad de tener una aplicación mucho más completa y pueda ser de gran ayuda a las personas.
- Implementar dentro del backend carpetas "test" y archivos spec con el fin de separar ya sean los controladores, módulos y servicios de prueba con los archivos de producción,

esto con la finalidad de que se puedan implementar librerías que faciliten las pruebas que se deseen realizar y que se puedan detectar de manera óptima, segura y rápida problemas si se presentasen.

7. ANEXOS

7.1. Entrevista para Levantamiento de Requerimientos

Nombre del entrevistado: Evelyn Peña

Nombre del entrevistador: Erick Molina

Cargo/Posición: Nutricionista

¿Quiénes serán los usuarios principales del sistema?

Nutricionistas

Pacientes

Administradores

Otros (especificar): _____

¿Qué características consideran esenciales para los usuarios nutricionistas?

Personalización de planes de dieta

Seguimiento de progreso de los pacientes

Comunicación con los pacientes

Generación de reportes nutricionales

Otros (especificar): _____

¿Qué características son esenciales para los usuarios pacientes?

Acceso a su plan de dieta

Registro diario de comidas

Registro de actividades físicas

Recepción de recordatorios/notificaciones

Comunicación con el nutricionista

Otros (especificar): _____

¿Qué tipo de datos necesita el sistema manejar y almacenar?

Información personal de los pacientes

Historial médico

Preferencias y restricciones alimentarias

Registros de comidas y actividades

Datos de progreso (peso, medidas, etc.)

Otros (especificar): _____

Nombre del entrevistado: Evelyn Peña

Nombre del entrevistador: Erick Molina

Cargo/Posición: Nutricionista

¿Quiénes serán los usuarios principales del sistema?

Nutricionistas

Pacientes

Administradores

Otros (especificar): _____

¿Qué características consideran esenciales para los usuarios nutricionistas?

Personalización de planes de dieta

Seguimiento de progreso de los pacientes

Comunicación con los pacientes

Generación de reportes nutricionales

Otros (especificar): _____

¿Qué características son esenciales para los usuarios pacientes?

Acceso a su plan de dieta

Registro diario de comidas

Registro de actividades físicas

Recepción de recordatorios/notificaciones

Comunicación con el nutricionista

Otros (especificar): _____

¿Qué tipo de datos necesita el sistema manejar y almacenar?

Información personal de los pacientes

Historial médico

Preferencias y restricciones alimentarias

Registros de comidas y actividades

Datos de progreso (peso, medidas, etc.)

Otros (especificar): _____

¿Qué aspectos considera importantes dentro de la aplicación?

Elegir preferencias alimenticias y la correcta generación de dietas

¿Cuántas comidas diarias se sugiere implementar para los usuarios?

Considerando los objetivos se considera 4 tomas diarias

¿Cuál sería el rango indicado de calorías consumidas diariamente para las personas que desean:

Bajar de peso: *H=1200-1800 / H=1500-2200*

Aumentar masa muscular: *H=2200-3250 / H=2500-3500*

Mantener su peso: *H=1800-2500 / H=2200-3000*

¿Cree y considera importante elegir los alimentos de preferencia antes de generar una dieta?

Sí

No

Comentarios adicionales:

Puede reaccionar a una alergia alimentaria

Contacto y Seguimiento

¿Está dispuesto a participar en futuras entrevistas o sesiones de feedback?

Sí

No

¿Hay algún otro miembro de su equipo o colegas que cree que deberíamos entrevistar?

Sí (Especificar contacto): _____

No

Firma Del Entrevistador

Erick Molina

Firma del Entrevistado

Evelyn Peña

8. BIBLIOGRAFÍA

Amazon Web Services, Inc. (s. f.-a). *Bases de datos no relacionales | Bases de datos de gráficos.*

Amazon Web Services, Inc. Recuperado 29 de mayo de 2024, de <https://aws.amazon.com/es/nosql/>

Amazon Web Services, Inc. (s. f.-b). *¿Qué es Flutter? - Explicación de la aplicación Flutter.*

Amazon Web Services, Inc. Recuperado 29 de mayo de 2024, de <https://aws.amazon.com/es/what-is/flutter/>

Amazon Web Services, Inc. (s. f.-c). *¿Qué es una API? - Explicación de interfaz de programación de aplicaciones - AWS.* Amazon Web Services, Inc. Recuperado 29 de mayo de 2024, de <https://aws.amazon.com/es/what-is/api/>

Amazon Web Services, Inc. (s. f.). *¿Qué es una base de datos? - Explicación de las bases de datos en la nube.* Amazon Web Services, Inc. Recuperado 29 de mayo de 2024, de <https://aws.amazon.com/es/what-is/database/>

APD. (2024, abril 9). *¿Qué es la metodología Scrum y cómo aplicarla? | APD.* APD España. <https://www.apd.es/metodologia-scrum-que-es/>

Atmitim, J. M. A., & Maluenda, R. (2021, febrero 22). Qué es Ionic: Ventajas y desventajas de usarlo en apps móviles híbridas. *Profile Software Services.* <https://profile.es/blog/que-es-ionic/>

Bautista García, I. J. (2021, marzo 30). *Backend y Frontend, ¿Qué es y cómo funcionan en la programación?* Servnet. <https://www.servnet.mx/blog/backend-y-frontend-partes-fundamentales-de-la-programacion-de-una-aplicacion-web>

Cargua Jimenez, K. L., & Fernandez Hidalgo, M. S. (2017). *“DESARROLLO DE UNA APLICACIÓN MÓVIL PARA EL CONTROL NUTRICIONAL Y DESARROLLO*

INFANTIL COMO UNA FORMA DE PREVENIR LA OBESIDAD” [ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL].

<https://www.dspace.espol.edu.ec/bitstream/123456789/44439/1/D-106409.pdf>

Deloitte Spain. (s. f.). *¿Qué es React Native?* Deloitte Spain. Recuperado 29 de mayo de 2024, de <https://www2.deloitte.com/es/es/pages/technology/articles/que-es-react-native.html>

Fis, E. (2023, septiembre 26). *¿Qué es el desarrollo móvil nativo?* Data IQ. <https://dataiq.com.ar/blog/desarrollo-movil-nativo/>

Gómez, P. (2022, diciembre 20). *¿Qué es el Backend?* *DevCamp*. <https://devcamp.es/que-es-backend/>

Hosting Plus. (2021, julio 6). *Modelo de prototipos: ¿qué es y cuáles son sus etapas?* *Hosting Plus*. <https://www.hostingplus.pe/blog/modelo-de-prototipos-que-es-y-cuales-son-sus-etapas/>

IBM. (s. f.). *Introducción al desarrollo de aplicaciones móviles | IBM*. IBM. Recuperado 29 de mayo de 2024, de <https://www.ibm.com/mx-es/topics/mobile-application-development>

Kinsta. (2022, julio 25). *¿Qué es Nest.js? Un Vistazo al Framework Ligero de JavaScript*. Kinsta. <https://kinsta.com/es/base-de-conocimiento/nestjs/>

López, M. (2023, abril 24). *Aplicaciones Híbridas, Claves en Desarrollo Mobile*. IMMUNE. <https://immune.institute/blog/aplicaciones-hibridas-desarrollo-mobile/>

Oracle. (s. f.-a). *¿Qué es MySQL?* Oracle. Recuperado 29 de mayo de 2024, de <https://www.oracle.com/co/mysql/what-is-mysql/>

Oracle. (s. f.-b). *¿Qué es una base de datos relacional?* Oracle. Recuperado 29 de mayo de 2024, de <https://www.oracle.com/cl/database/what-is-a-relational-database/>

- Organización Mundial de la Salud. (2019, agosto 25). *Dieta sana*. Organización Mundial de la Salud. <https://www.who.int/es/health-topics/healthy-diet>
- Raeburn, A. (2024, febrero 13). *¿Qué es la programación extrema (XP)?* Asana. <https://asana.com/es/resources/extreme-programming-xp>
- Rommel, F. (s. f.). *SQLite: La Base de Datos Embebida*. SG Buzz. Recuperado 29 de mayo de 2024, de <https://sg.com.mx/revista/17/sqlite-la-base-datos-embebida>
- Santacruz Ayala, A. C., & Villegas Proaño, W. E. (2020). *DESARROLLO DE APLICACIÓN MÓVIL PARA PLANIFICACIÓN DE DIETAS SALUDABLES* [ESCUELA POLITÉCNICA NACIONAL]. <https://bibdigital.epn.edu.ec/bitstream/15000/21200/1/CD%2010716.pdf>
- ServiceNow. (s. f.). *¿Qué es el desarrollo móvil?* ServiceNow. Recuperado 29 de mayo de 2024, de <https://www.servicenow.com/es/workflows/creator-workflows/what-is-mobile-development.html>
- Software DELSOL. (2019, abril 1). ▷ *Metodología ¿Qué es?* Software DELSOL. <https://www.sdelzol.com/glosario/metodologia/>
- Tecnologías Información. (s. f.). *Modelos de datos: Modelo Conceptual, Físico y Lógico*. Tecnologías Información. Recuperado 29 de mayo de 2024, de <https://www.tecnologias-informacion.com/modelos-datos.html>
- The Texas Heart Institute. (2019, noviembre 15). *Nutrición*. The Texas Heart Institute. <https://www.texasheart.org/heart-health/heart-information-center/topics/nutricion/>