

MYFOODPLANNER: APP PARA RECOMENDACIONES DE COMIDAS SALUDABLES



TRABAJO FIN DE GRADO
CURSO 2023-2024

AUTORES

ANTONIO BONILLO SAIZ ¹
BLANCA GARCÍA ÁLVAREZ ²
VÍCTOR ORTEGO MARTÍN ²
JOSÉ ÁNGEL SUÁREZ PESQUERA ²

DIRECTOR

PABLO RABANAL BASALO

GRADO EN INGENIERÍA DE COMPUTADORES ¹.
GRADO EN INGENIERÍA DE SOFTWARE ².
FACULTAD DE INFORMÁTICA
UNIVERSIDAD COMPLUTENSE DE MADRID

MYFOODPLANNER: HEALTHY FOOD RECOMMENDATIONS APP



TRABAJO DE FIN DE GRADO EN INGENIERÍA DE COMPUTADORES.
TRABAJO DE FIN DE GRADO EN INGENIERÍA DEL SOFTWARE.

AUTORES

ANTONIO BONILLO SAIZ ¹
BLANCA GARCÍA ÁLVAREZ ²
VÍCTOR ORTEGO MARTÍN ²
JOSÉ ÁNGEL SUÁREZ PESQUERA ²

DIRECTOR

PABLO RABANAL BASALO

CONVOCATORIA: JUNIO 2024

GRADO EN INGENIERÍA DE COMPUTADORES ¹.
GRADO EN INGENIERÍA DE SOFTWARE ².
FACULTAD DE INFORMÁTICA
UNIVERSIDAD COMPLUTENSE DE MADRID

27 DE MAYO DE 2024

AGRADECIMIENTOS

Agradecimientos de Antonio Bonillo Saiz

Antes que a nadie me gustaría agradecerle a mi familia su apoyo, han sido los primeros a los que les he enseñado los progresos de mis proyectos, y aunque muchas veces no entendiesen de qué les hablaba, siempre me han escuchado.

También quiero agradecer a mis amigos que han sabido levantarme los ánimos cuando lo necesitaba, en especial a Álvaro que ha sido el que me ha hecho ver siempre las cosas con paciencia y perspectiva.

Por último agradecer a todas las personas que en algún momento a lo largo de la carrera han hecho por ayudarme. Espero haber estado a la altura y haberles ayudado como mínimo lo mismo que me han ayudado a mí.

Agradecimientos de Blanca García Alvarez

Quisiera expresar mi más sincero agradecimiento a todas las personas que han sido parte esencial de esta trayectoria académica. Primero de todo quiero dar las gracias a mi familia, por su inquebrantable apoyo y entendimiento. Sus palabras de ánimo, especialmente la famosa frase y muy escuchada en mi casa "es una carrera de fondo", han sido un pilar fundamental para mantenerme firme y dar siempre lo mejor de mí.

A mi novio, gracias por toda la ayuda y el apoyo que me has brindado a lo largo de estos siete años de carrera. Tu paciencia, comprensión y amor han sido indispensables para superar los desafíos y alcanzar este logro.

No puedo olvidar a las personas y profesores que, de una u otra manera, han facilitado este largo y complicado proceso.

Finalmente, un agradecimiento especial a aquellos que no confiaban, ya que han sido una fuente adicional de motivación para demostrar de lo que soy capaz.

Agradecimientos de Víctor Ortego Martín

Quiero dar las gracias primero de todo a mi familia, cuyo apoyo ha sido muy importante para mí durante estos años de carrera.

Agradecer a mis compañeros de proyecto por su dedicación y colaboración durante todos estos meses, ha sido un camino largo.

También quiero reconocer el aporte de los profesores y el personal de la universidad, cuya enseñanza y orientación han sido fundamentales en mi desarrollo académico. Finalmente, gracias a todas las personas que han formado parte de etapa universitaria.

Agradecimientos de José Ángel Suárez Pesquera

Quiero agradecer a mi familia, quienes han sido mi pilar fundamental durante este duro viaje académico. Su apoyo incondicional, comprensión y amor han sido mi inspiración constante. Gracias por creer en mí y por alentarme en cada paso del camino. Este logro no sería posible sin su sacrificio y aliento constante.

Agradecer a mi novia, por su apoyo incondicional que ha supuesto mucho sacrificio entre los dos pero siempre ha estado ahí para cualquier cosa que necesitara y animándome para continuar. Sin ella no hubiera sido capaz de alcanzar esta meta.

Por otro lado, a mis amigos, por estar a mi lado en los momentos buenos y malos. Vuestras palabras de ánimo, risas compartidas y momentos de distracción han sido mi salvavidas en medio de la intensidad de este proceso.

A todos aquellos que de una forma u otra han contribuido a este proyecto, vuestro apoyo y orientación han sido de gran ayuda. A mis profesores, gracias por compartir vuestro conocimiento y experiencia. Este logro no solo es mío, sino también de todos ustedes, que han sido parte de este viaje.

RESUMEN

MYFOODPLANNER. APP PARA RECOMENDACIONES DE COMIDA SALUDABLE

El presente trabajo consiste en el desarrollo de una aplicación Android para recomendaciones de comida saludable. Este proyecto tiene como objetivo principal la creación de una herramienta que genere menús diarios equilibrados para desayuno, comida y cena, promoviendo hábitos alimenticios saludables. Para ello, la interfaz de usuario será intuitiva y amigable, que permita una fácil navegación por la aplicación.

La aplicación permitirá a los usuarios personalizar sus preferencias calóricas, ya sea introduciendo el número de calorías diarias deseadas o calculando automáticamente las calorías recomendadas según datos individuales como su peso, altura, edad, sexo y hábitos de ejercicio.

Esta aplicación busca fomentar un estilo de vida saludable al ofrecer opciones de alimentación adaptadas a las necesidades específicas de cada usuario. Además, se centrará en la diversidad de alimentos, teniendo en cuenta criterios nutricionales para garantizar la ingesta adecuada de nutrientes esenciales.

Palabras clave: aplicación móvil android, dietas, menús, comida saludable, seguimiento de peso.

ABSTRACT

MYFOODPLANNER. HEALTHY FOOD RECOMMENDATIONS APP

The present work consists of an Android application for healthy food recommendations, aiming to create a tool that generates balanced weekly menus for breakfast, lunch and dinner, promoting healthy eating habits. The user interface will be intuitive and user-friendly, allowing easy navigation through the application.

The application will enable users to customize their calorie preferences, either by entering the desired daily calorie count or by automatically calculating recommended calories based on individual data such as weight, height, age, gender, and exercise habits.

This application seeks to promote a healthy lifestyle by providing food options tailored to the specific needs of each user. Furthermore, it will focus on food diversity, considering nutritional criteria to ensure the adequate intake of essential nutrients.

Keywords: android mobile app, diets, menus, healthy food, weight monitoring.

ÍNDICE DE CONTENIDOS

Capítulo 1: Introducción	14
1.1. Motivación	15
1.2. Objetivos del TFG	16
1.3. Plan de trabajo	18
Capítulo 2: Estado de la cuestión	20
2.1. Aplicaciones similares	20
2.1.1. FoodData Central	21
2.1.2. PlanEat	21
2.1.3. 8FIT	22
2.1.4. MyFitnessPal	22
2.1.5. Halen	23
2.1.6. MyFoodPlan	23
2.2. Application Programming Interface (API) de recetas	24
2.2.1. Apilayer	24
2.2.2. Spoonacular Application Programming Interface	25
2.2.3. Nutritionix	25
2.2.4. Nutritics Application Programming Interface	26
2.2.5. ReciPal	26
2.2.6. FatSecret Platform Application Programming Interface	27
2.2.7. LogMeal Application Programming Interface	27
2.2.8. Chomphthis	28
2.2.9. BigOven Application Programming Interface	28
2.2.10. TheMealDB	28
2.2.11. Zestful	29
2.3. Indicaciones y referencias alimenticias	30
2.3.1. Tasa Metabólica Basal	30
2.3.2. Adaptación de TMB a ejercicio físico semanal	31
2.3.3. Macronutrientes	31
2.3.4. Distribución de macronutrientes	33
Capítulo 3: Tecnologías para el desarrollo de la aplicación	34
3.1. Entorno de desarrollo: Android Studio	34
3.2. Lenguaje de programación: Java	35

3.3. API Spoonacular	35
3.4. Herramientas de apoyo	38
3.4.1. Image Creator from Microsoft Designer	38
3.4.2. Git y Github	38
3.4.3. DB Browser for SQLite	38
3.4.4. Google Fonts	39
Capítulo 4: Casos de uso	40
4.1. Gestión de usuarios	41
4.1.1. Alta usuario	41
4.1.2. Modificación de perfil de usuario	44
4.1.3. Eliminación de perfil de usuario	46
4.1.4. Cierre de sesión en la aplicación	48
4.2. Gestión de Dietas	50
4.2.1. Creación de una dieta	50
4.2.2. Eliminación de dieta	52
4.2.3. Cambio de menú diario	54
4.3. Seguimiento de peso	56
4.3.1. Registro de peso de un usuario	57
4.3.2. Modificación de peso de un usuario	58
4.3.3. Consultar gráfica de peso	60
4.3.4. Añadir peso al gráfico	61
4.4. Gestión de la lista de la compra	63
Capítulo 5: Descripción de la aplicación desarrollada	64
5.1. Administración de perfiles	65
5.2. Pantalla principal y navegación	67
5.3. Generación de una dieta	69
5.4. Visualización del menú diario	72
5.5. Lista de la compra	76
5.6. Seguimiento de peso	78
5.7. Evolución óptima de peso	79
Capítulo 6: Detalles de implementación	80
6.1. Modelo de datos	80
6.2. Detalles de implementación	84
Capítulo 7: Test en usuarios	88

7.1. Generación de la APK	88
7.2. Funcionalidades evaluadas	89
7.3. Resultado de la evaluación	91
Capítulo 8: Conclusiones y trabajo futuro	99
8.1. Conclusiones	99
8.2. Trabajo Futuro	100
Capítulo 9: Conclusions and future work	101
9.1. Conclusions	101
9.2. Future Work	102
Contribuciones personales	103
Contribuciones de Antonio Bonillo Saiz	103
Contribuciones de Blanca García Álvarez	106
Contribuciones de Víctor Ortego Martín	109
Contribuciones de Jose Ángel Suárez Pesquera	112
Bibliografía	115

ÍNDICE DE FIGURAS

2.1. FoodDataCentral	21
2.2. PlanEat	21
2.3. 8Fit	22
2.4. MyFitnessPal	22
2.5. Halen	23
2.6. MyFoodPlan	23
2.7. Apilayer	24
2.8. Spoonacular API	25
2.9. Nutritionix	25
2.10. Nutritics API	26
2.11. ReciPal	26
2.12. FatSecret Platform API	27
2.13. LogMeal API	27
2.14. Chompthis	28
2.15. BigOven API	28
2.16. TheMealDB	28
2.17. Zestful	29
3.1. Iconos de ejemplo	39

4.1. Diagrama UML de gestión de usuarios	41
4.2. Diagrama UML de gestión de dietas	50
4.3. Diagrama UML de seguimiento de peso	56
5.1 Logo de la aplicación	64
5.2. Pantalla acceso a usuarios	65
5.3. Pantalla crear un usuario	66
5.4: Pantalla borrado de un perfil	67
5.5: Pantalla principal de la aplicación	68
5.6. Menú desplegable de navegación	69
5.7. Pantalla de introducción de datos	70
5.8. Pantalla de opciones de creación de dieta	71
5.9. Pantalla de introducción de tipo de dieta	72
5.10. Calendario de dieta	73
5.11. Visualización de un menú diario	74
5.12. Visualización de un plato individual	75
5.13. Pantalla de selección de ingredientes	77
5.14. Pantalla principal de visualización de progresión	78
5.15. Visualización de serie de peso óptimo	79
6.1. Modelo entidad-relación	83

6.2. Petición a la API para generar una dieta	84
6.3. Resultado de generar una dieta un menú de un día	85
6.4. Petición a la API para obtener información de una receta	86
6.5. Petición a la API para autocompletar ingredientes	86
6.6. Resultado de petición autocompletar un ingrediente	87
7.1. Resultados de la pregunta 1	91
7.2. Resultados de la pregunta 2	91
7.3. Resultados de la pregunta 3	92
7.4. Resultados de la pregunta 4	93
7.5. Resultados de la pregunta 5	93
7.6. Resultados de la pregunta 6	94
7.7. Resultados de la pregunta 7	95
7.8. Resultados de la pregunta 8	95
7.9. Resultados de la pregunta 9	96
7.10. Resultados de la pregunta 10	97
7.11. Resultados de la pregunta 11	97
7.12. Resultados de la pregunta 12	98

ÍNDICE DE TABLAS

1.1. Planificación del proyecto	18
2.1. Estimación de la TMB	30
2.2. Cálculo de calorías diarias en base a actividad física semanal	31
2.3. Distribución de macronutrientes por sexo	33
6.1. Tabla <i>t_usuarios</i>	80
6.2. Tabla <i>t_dietas</i>	81
6.3. Tabla <i>t_dietmeals</i>	81
6.4. Tabla <i>t_meal</i>	81
6.5. Tabla <i>t_ingredientsmeals</i>	82
6.6. Tabla <i>t_weightmonitoring</i>	82
6.7. Tabla <i>t_advices</i>	82

Capítulo 1: Introducción

Actualmente la tecnología desempeña un papel fundamental en nuestro día a día, la salud y la nutrición encuentran un aliado poderoso en las aplicaciones móviles. En este contexto, el presente Trabajo de Fin de Grado se sumerge en el fascinante mundo de la ingeniería de software y la nutrición, combinando ambos para desarrollar una aplicación Android innovadora y práctica. Nuestra propuesta se centra en la creación de una aplicación dedicada a la generación personalizada de dietas, con el objetivo de brindar a los usuarios una herramienta accesible y efectiva para mejorar sus hábitos alimenticios.

A lo largo de este trabajo, se explorará el proceso de desarrollo de la aplicación, desde la concepción de la idea hasta la implementación práctica. Se abordarán las consideraciones técnicas, las decisiones de diseño y las funcionalidades clave que definen esta herramienta digital única. Además, se evaluará la efectividad y la usabilidad de la aplicación a través de estudios de casos y retroalimentación de usuarios, con el propósito de validar su potencial impacto positivo en la promoción de estilos de vida más saludables.

Con este proyecto, no solo buscamos contribuir al campo de la ingeniería de software, sino también proporcionar una herramienta valiosa para aquellos que buscan mejorar su salud a través de elecciones dietéticas informadas. La convergencia entre la tecnología y la nutrición se manifiesta en esta aplicación Android, que aspira a convertirse en un compañero confiable para quienes buscan un cambio positivo en sus hábitos alimenticios.

1.1. Motivación

En la era moderna, donde la rapidez y la conveniencia a menudo priman en nuestra rutina diaria, la alimentación saludable se ha convertido en un desafío crucial. El sobrepeso y la obesidad han alcanzado proporciones alarmantes, afectando la calidad de vida y la salud de millones de personas en todo el mundo. Este problema no solo tiene implicaciones físicas, sino que también impacta en la salud mental y en la prevalencia de enfermedades crónicas. La obesidad y el sobrepeso conforman el quinto factor de riesgo de muerte en el mundo, y es que en el caso de los españoles, la prevalencia de sobrepeso en la población adulta en el año 2012 fue del 39,0% (47,3% en hombres y 31,1% en mujeres) y la de obesidad del 18,5% (19,3% en hombres y 17,8% en mujeres). [1].

Enfrentar el sobrepeso no solo es una cuestión estética, sino una necesidad imperante para preservar la salud a largo plazo. La carga adicional de peso no solo ejerce presión sobre las articulaciones y órganos internos, sino que también desencadena una cascada de efectos adversos en el sistema metabólico.

La preocupación por abordar este desafío ha llevado a la aparición de métodos y corrientes que buscan educar y concienciar sobre los peligros de una mala alimentación, así como distintas maneras de facilitar y acompañar el proceso de mantener una vida y alimentación saludables.

En este contexto, exploraremos una aplicación específica diseñada para promover una alimentación saludable, ofreciendo soluciones prácticas y accesibles para usuarios que pueden no estar completamente familiarizados con conceptos nutricionales.

1.2. Objetivos del TFG

El objetivo de este proyecto es el desarrollo de una aplicación para la creación y seguimiento de dietas personalizadas en función de las necesidades del usuario, que podrá generar una dieta de la duración que elija, en función de sus características fisiológicas y particularidades personales. Para ello es crucial que el usuario pueda introducir sexo, altura y edad, además de poder elegir el tipo de dieta que quiere seguir y darle la posibilidad de excluir ciertos alimentos que no son de su preferencia.

La aplicación propuesta tiene como objetivo principal ofrecer una experiencia eficiente en la gestión de la salud y la nutrición de los usuarios. En primer lugar, permitirá la creación y gestión de múltiples perfiles de usuario en una sola aplicación, brindando así una solución personalizada para cada individuo. Además, la aplicación calculará las necesidades calóricas específicas de cada usuario, considerando sus características fisiológicas para proporcionar recomendaciones nutricionales precisas.

Uno de los aspectos destacados que la aplicación perseguirá, será la flexibilidad para que los usuarios puedan excluir alimentos de su dieta según sus preferencias y necesidades. Además, presentará la posibilidad de elegir entre diferentes tipos de dietas, como vegetariana, vegana o baja en carbohidratos, entre otras, proporcionando así opciones adaptadas a diversos estilos de vida y preferencias alimenticias.

La funcionalidad clave de la aplicación residirá en la generación de dietas personalizadas, integrando la información recopilada anteriormente. Esto incluye la capacidad de utilizar un API que proporciona recetas y poder combinarlas para formar un menú completo durante un período específico de

tiempo. Cada receta generada incluirá información detallada sobre ingredientes y métodos de elaboración, facilitando la preparación y seguimiento de las comidas.

Además, la aplicación simplificará la planificación de compras al generar y almacenar una lista de la compra que contendrá todos los ingredientes necesarios para seguir la dieta durante el periodo seleccionado. Asimismo, se centrará en la gestión y visualización eficiente de las recetas diarias, permitiendo a los usuarios explorar y seguir fácilmente el plan alimenticio establecido.

Otro componente importante de la aplicación será la capacidad de introducir, almacenar y visualizar el peso del usuario diariamente. Esto permitirá el seguimiento continuo del progreso y la generación de gráficas que mostrarán la evolución del peso a lo largo del tiempo. Además, se incorporará la funcionalidad de generar series de datos sobre el peso óptimo en función de las características específicas de la dieta del usuario, proporcionando una perspectiva valiosa para la toma de decisiones informadas relacionadas con la salud y la nutrición. En conjunto, estos propósitos consolidan la aplicación como una herramienta completa y personalizada para la mejora de los hábitos alimenticios y el seguimiento del peso.

Por último, se mostrarán consejos al usuario cuya idea es motivar para seguir con el mismo esfuerzo y la misma dedicación de su plan.

1.3. Plan de trabajo

Con el fin de conseguir los objetivos enumerados en el apartado anterior, se diseñó un plan de trabajo, que se muestra en la tabla 1.1, a continuación:

<u>Objetivo</u>	<u>Periodo de tiempo</u>	<u>Descripción</u>
1. Análisis de aplicaciones similares	Septiembre de 2023. Dos semanas.	Buscar y estudiar aplicaciones que tengan funcionalidades semejantes a las que queremos implementar.
2. Búsqueda de herramientas para la generación de menús	Septiembre de 2023. Dos semanas.	Recopilar elementos que nos permitan especificar platos así como sus ingredientes y recetas. Descartar aquellos que su acceso y uso no sea gratis. Comprobar que sus funcionalidades sean suficientes para satisfacer los requisitos del proyecto.
3. Estudio de las tecnologías a usar	Octubre de 2023. Dos semanas.	Realizar un estudio de las tecnologías que se iban a utilizar para el desarrollo de la aplicación, así como la estructura de la base de datos.
4. Especificación de funcionalidades a implementar	Octubre de 2023. Una semana.	Detallar el coste de implementación y enumerar las características que permiten que la funcionalidad se comporte como está previsto.
5. Diseño de la aplicación	Última semana de octubre de 2023 hasta diciembre de 2024.	Diseñar la arquitectura de la aplicación, de la base de datos y la interfaz de usuario de la aplicación.

6. Desarrollo e implementación	Desde diciembre de 2023 hasta marzo de 2024.	Desarrollar el software de acuerdo con las especificaciones establecidas durante la etapa de diseño y llevar a cabo su implementación.
7. Realización de pruebas y corrección de errores.	Abril de 2024. Dos semanas.	Mediante pruebas y correcciones, hemos identificado y solucionado los defectos, asegurando que el producto final cumpla con los requisitos establecidos.
8. Documentación	Dos últimas semanas de abril y dos primeras de mayo de 2024.	Elaboración de la memoria del proyecto como soporte a la aplicación.

Tabla 1.1: Planificación del proyecto

Capítulo 2: Estado de la cuestión

En esta revisión del estado de la cuestión, se explorarán las características y limitaciones de las aplicaciones existentes, destacando la necesidad de una solución integral que aborde de manera efectiva las diversas dimensiones de la gestión dietética personalizada. Además, se analizarán tendencias emergentes en el ámbito de la tecnología aplicada a la salud, proporcionando un contexto relevante para comprender la importancia y la mejora de nuestra propuesta. Con esta base, el proyecto se propone contribuir significativamente al campo, ofreciendo una herramienta que no solo se adapte a las necesidades actuales, sino que también se anticipe a las futuras demandas de un público cada vez más comprometido con la mejora de sus hábitos alimenticios y su bienestar general.

2.1. Aplicaciones similares

Antes de realizar el diseño y el desarrollo de la aplicación, se realizó una búsqueda en el mercado Android sobre aplicaciones con funcionalidades similares a las que se quieren implementar. A continuación, se muestran algunas aplicaciones que se probaron:

2.1.1. FoodData Central

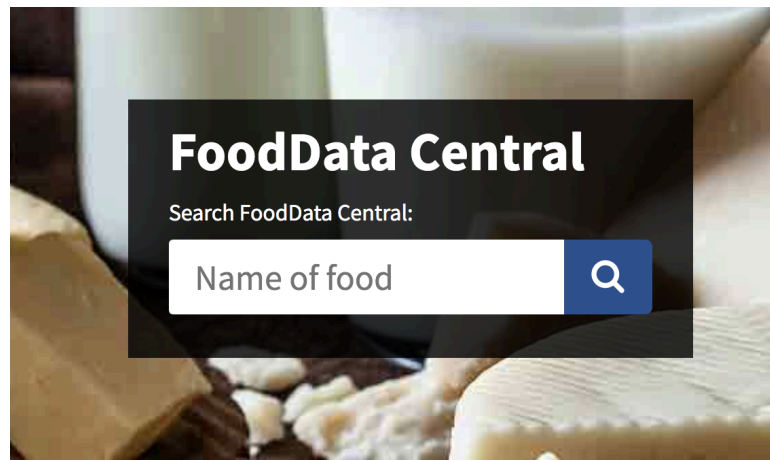


Figura 2.1: [2]

Es una plataforma que recopila información nutricional sobre alimentos. No genera dietas pero muestra por cada alimento sus valores nutricionales. Puede ser útil para proporcionar más información al usuario sobre los alimentos que consume. Esta recopilación de información es gratis.

2.1.2. PlanEat



Figura 2.2: [3]

Es una aplicación que mediante diferentes datos personales del usuario realiza menús semanales, lista de compra semanal, diferentes entrenos gratuitos, escáner de

productos y tutoriales de recetas. Una aplicación que contiene algunas funcionalidades similares a la idea principal del proyecto.

2.1.3. 8FIT



Figura 2.3: [4]

Esta aplicación se caracteriza por contener gran número de recetas en las que fijarse para realizar sus propios menús. Cada receta contiene fotos de la misma además de sus diferentes valores nutricionales y calificativos. Y como complemento, permite acceder a diversos entrenamientos con sus guías correspondientes.

2.1.4. MyFitnessPal



Figura 2.4: [5]

Es una aplicación para realizar un conteo de calorías y macros en las diferentes comidas del día. No genera menús, pero puedes incluir en ella las diferentes comidas ingeridas. Sirve como informativo para saber si se está cumpliendo con el objetivo.

2.1.5. Halen



Figura 2.5: [6]

En esta aplicación, mediante diferentes datos personales del usuario, proporciona una dieta con diferentes recetas de nutricionistas profesionales. Dentro de cada receta incluye sus ingredientes, instrucciones para cocinarlo y sus valores nutricionales. Además, permite añadir o cambiar dichas recetas por otras y guardarlas en favoritos. Y con las recetas asignadas por la aplicación, se genera automáticamente una lista de la compra.

2.1.6. MyFoodPlan



Figura 2.6: [7]

En ella permite indicar sus datos personales y el objetivo principal que tiene con la aplicación. Con esto, se genera una dieta con diferentes recetas con sus respectivos valores nutricionales e instrucciones. La novedad de esta aplicación es que contiene un blog donde se pueden encontrar numerosos artículos sobre comidas saludables, alimentos y retos. Y además, incluye recomendaciones generales como la cantidad

diaria de agua que hay que tomar o la realización diaria de deporte para mantener una vida saludable.

2.2. Application Programming Interface (API) de recetas

Como la aplicación está relacionada con la generación de dietas con diferentes menús y recetas, se ha decidido utilizar una API o base de datos sobre recetas y alimentos para su obtención. Por lo que se llevó a cabo una búsqueda de diferentes API's que proporcionan recetas, con sus ingredientes necesarios y los debidos pasos para realizarla. A continuación, se muestran algunas API que se investigaron:

2.2.1. Apilayer



Figura 2.7: [8]

Es una base de datos nutricional con alrededor de 900.000 artículos que proporciona datos detallados sobre los valores nutricionales de sus alimentos. Permite búsquedas por nombre, código de barras o palabras clave. Sin embargo, su número de solicitudes por minuto es muy limitado y para aumentar dicha restricción se requiere un plan de pago por 19\$ al mes.

2.2.2. Spoonacular Application Programming Interface



Figura 2.8: [9]

Spoonacular cuenta con una amplia gama de datos que incluyen 2.600 ingredientes, 5.000 recetas, información de precios y consejos de cocina. Además, permite generar menús diarios o semanales según diferentes restricciones. Ofrece un plan gratuito con 150 solicitudes diarias, además de otros planes de pago con características adicionales.

2.2.3. Nutritionix



Figura 2.9: [10]

Nutritionix ofrece una base de datos con más de 800.000 ingredientes y 184.000 comidas de diferentes restaurantes. Permite buscar restaurantes por comidas o recetas. Tiene un plan gratuito muy limitado y varios planes de pago que varían según las características que ofrecen.

2.2.4. Nutritics Application Programming Interface



Figura 2.10: [11]

Es una API bastante similar a la anterior, pero en vez de tener recetas o menús de restaurantes, esta base de datos contiene recetas de diferentes marcas. Su uso es exclusivo para usuarios con planes de pago, los cuales son muy similares de precio, pero varían según el objetivo que quieras sacar a la API, desde utilizarla para nutrición y salud, para deportistas o incluso para servicio de alimentos.

2.2.5. ReciPal



Figura 2.11: [12]

ReciPal más que una base de datos sobre recetas, se enfoca más en la gestión de recetas y etiquetas nutricionales. Permite almacenar recetas en la nube con su información nutricional y sus precios, lo cual puede ser interesante como complemento para los usuarios.

2.2.6. FatSecret Platform Application Programming Interface



Figura 2.12: [13]

Esta API alberga alrededor de 1 millón de artículos que incluyen productos de marca, artículos de restaurantes y alimentos genéricos. Ofrece un plan gratuito limitado a 5.000 llamadas diarias, así como planes de pago con características adicionales. Una cosa interesante es que para estudiantes u organizaciones sin fines de lucro, ofrecen un plan de pago de forma gratuita.

2.2.7. LogMeal Application Programming Interface



Figura 2.13: [14]

LogMeal API se caracteriza por proporcionar información detallada sobre nutrientes de alimentos y tiene capacidades de reconocimiento de platos y tipos de alimentos. Ofrece una prueba gratuita de 30 días y posteriormente diferentes planes de pago con diferentes características.

2.2.8. Chomphthis



Figura 2.14: [15]

Base de datos con alrededor de 875.000 alimentos de diferentes marcas, ingredientes y productos de varios países con información detallada. Tiene un plan gratuito muy limitado, donde cada solicitud tiene el coste de 1 céntimo, y como suplemento, diferentes planes de pago.

2.2.9. BigOven Application Programming Interface



Figura 2.15: [16]

BigOven ofrece funciones de búsqueda avanzada a sus productos, creación de listas de compras y permite agregar recetas y contenido personalizado a sitios web de terceros. Pero su utilización es solamente mediante planes de pago que empiezan desde los \$99 al mes.

2.2.10. TheMealDB



Figura 2.16: [17]

TheMealDB es una base de datos en desarrollo con un número muy limitado de comidas, ingredientes e imágenes de los mismos. Su utilización es gratuita, pero para poder agregar recetas y realizar búsquedas avanzadas en la base de datos se necesita pagar una tarifa adicional.

2.2.11. Zestful



Figura 2.17: [18]

Zestful es una base de datos que permite búsquedas fáciles a productos y la creación de listas de compras. Además, transforma los datos de los productos en formato JSON para facilitar su uso. Solamente permite 30 búsquedas diarias y cobra \$0.02 por ingrediente a empresas.

2.3. Indicaciones y referencias alimenticias

En esta sección, se van a explicar y detallar diferentes términos utilizados para la estimación de las kilocalorías, para así poder crear una dieta personalizada para cada usuario.

2.3.1. Tasa Metabólica Basal

Con el fin de estimar el gasto calórico de una persona, necesitamos una fórmula matemática que nos permita calcular este gasto en base a sexo, edad, altura y ejercicio físico.

La tasa metabólica basal (TMB) es el valor mínimo de energía que necesita una persona para subsistir en reposo. Esta energía se mide en kilocalorías y se destina al correcto funcionamiento de las constantes vitales de manera involuntaria así como del mantenimiento de una temperatura corporal adecuada y constante. [19]

Para realizar este cálculo se necesitan esencialmente 3 datos que ya hemos mencionado; sexo, edad y altura. A continuación se puede observar en la tabla 3.1 cómo se realiza este cálculo en base a la fórmula correspondiente.

<u>Sexo</u>	<u>TMB</u>
Hombre	$(10 \times \text{peso en kg}) + (6,25 \times \text{altura en cm}) - (5 \times \text{edad en años}) - 161$
Mujer	$(10 \times \text{peso en kg}) + (6,25 \times \text{altura en cm}) - (5 \times \text{edad en años}) - 161$

Tabla 2.1: Estimación de la TMB

2.3.2. Adaptación de TMB a ejercicio físico semanal

El ejercicio físico semanal es un factor determinante en el cálculo del gasto de energía diario, por lo tanto es necesario tenerlo en cuenta y añadir este gasto extra a la TMB que actúa como base. Entendemos ejercicio extra semanal como por ejemplo, la práctica de algún deporte, las actividades cotidianas como ir a trabajar, caminar etc entran dentro de poco o ningún ejercicio extra. A continuación se puede observar en la tabla 3.2 cómo se realiza este cálculo en base a la TMB y ejercicio físico semanal. [19]

<u>Ejercicio semanal</u>	<u>Calorías diarias necesarias</u>
Poco o ningún ejercicio extra	TMB x 1,2
Ejercicio extra 1-2 veces por semana	TMB x 1,375
Ejercicio extra 3-5 veces por semana	TMB x 1,55
Ejercicio extra 6-7 veces por semana	TMB x 1,725

Tabla 2.2: Cálculo de calorías diarias en base a actividad física semanal

2.3.3. Macronutrientes

El consumo y distribución de macronutrientes (proteínas, carbohidratos y grasas) es también un factor crucial desde el punto de vista nutricional. Estos nos proporcionan calorías que nuestro organismo degrada y utiliza como combustible para su correcto funcionamiento. A continuación detallamos brevemente las características particulares de cada macronutriente:

- **Proteínas**: proporcionan 4 kcal de energía por gramo. Están compuestas por aminoácidos que se pueden clasificar en esenciales y no esenciales.

“Los aminoácidos esenciales no pueden sintetizarse adecuadamente por el organismo para cubrir sus necesidades, por lo que deben aportarse a través de la dieta” [20]

Esto implica que se deben priorizar aquellos alimentos que contengan todos los aminoácidos esenciales, como son la carne, los huevos y los lácteos entre otros. Sin embargo, es posible combinar fuentes de ciertos aminoácidos que no contengan todos los aminoácidos esenciales de manera que el aporte total de aminoácidos sea el mismo que con las fuentes nombradas anteriormente. [20]

- **Grasas:** también conocidas como lípidos, proporcionan 9 kcal de energía por gramo. Es un macronutriente esencial para el correcto funcionamiento del organismo, ya que ayuda en la absorción de vitaminas, así como en la producción de hormonas. Se encuentran en alimentos como el pescado azul, las semillas y los frutos secos.

Se clasifican en ácidos grasos saturados e insaturados. Dentro de los insaturados, se encuentran los poliinsaturados, estos son especialmente importantes ya que el organismo carece de las enzimas necesarias para sintetizarlos, por lo que son componentes esenciales en la alimentación. [21]

- **Carbohidratos:** proporcionan 9 kcal de energía por gramo. Son compuestos derivados de las plantas que se clasifican en monosacáridos y polisacáridos, sin embargo para que estos puedan ser utilizados por el organismo, deben descomponerse en monosacáridos.

Son la fuente principal de energía para el metabolismo, entre los más importantes podemos encontrar la fructosa, la glucosa y la galactosa, presentes en frutas, verduras, legumbres y frutos secos entre otros. [22]

2.3.4. Distribución de macronutrientes

Una vez estimada la cantidad de energía necesaria en el día a día, debemos elegir en qué porcentajes se distribuyen los macronutrientes de manera que el cómputo total satisfaga las necesidades metabólicas del organismo así como las energéticas. Estas necesidades varían según el objetivo y fisiología del usuario, pero se pueden resumir en intervalos que indican tanto el mínimo como el máximo recomendado. A continuación se muestra la tabla 3.3 con los intervalos por sexo. [20]

<u>Sexo</u>	<u>Proteínas</u>	<u>Grasas</u>
Hombre	1,7 - 2,5 g/kg	0,7 - 1,2 g/kg
Mujer	1,6 - 1,8 g/kg	1 - 1,5 g/kg

Tabla 2.3: Distribución de macronutrientes por sexo

En esta tabla podemos observar que las distribuciones están muy diferenciadas, y es que, en el caso de las mujeres, necesitan una cantidad de grasa mayor ya que su cuerpo realiza funciones específicas que los hombres no, como por ejemplo la menstruación, el desarrollo fetal y la lactancia.

En el caso de los hombres, la cantidad de proteínas es mayor dado que por lo general, tienen un mayor componente de masa muscular en su cuerpo.

Ya que los carbohidratos no son imprescindibles para el cuerpo humano, no se distribuyen con un coeficiente determinado, sino que se utiliza para regular el superávit/ déficit según la dieta del usuario.

Por ejemplo, si el usuario pretende subir de peso, añadiremos carbohidratos de manera que sobrepase su necesidad de energía diaria. Sin embargo, si lo que pretende es bajar de peso, ajustaremos estos carbohidratos de manera que la energía no llegue al gasto calórico diario, y así generamos un déficit calórico. [20]

Capítulo 3: Tecnologías para el desarrollo de la aplicación

En el transcurso de este Trabajo de Fin de Grado (TFG), se ha llevado a cabo un gran estudio en el que la ingeniería de software y la tecnología convergen para dar origen a una aplicación muy completa y funcional. La clave de nuestro proyecto reside en la cuidadosa selección y aplicación de tecnologías que han desempeñado un papel fundamental en cada etapa del proceso de desarrollo. Esta introducción sirve como un breve vistazo a las herramientas y plataformas clave que han sido empleadas, delineando su importancia en la creación de una solución integral y efectiva.

3.1. Entorno de desarrollo: Android Studio

Android Studio [\[23\]](#) ha sido diseñado específicamente para el desarrollo de aplicaciones en Android, por ello es el entorno de desarrollo integrado oficial para dicha plataforma. Presenta varias ventajas que lo convierten en una elección popular entre los desarrolladores de Android, entre ellas podemos mencionar:

- Interfaz de usuario muy intuitiva y fácil de usar
- Compatibilidad con Kotlin y Java: es compatible tanto con Java como con Kotlin, lo que brinda a los desarrolladores la flexibilidad de elegir el lenguaje que prefieran para programar sus aplicaciones
- Emuladores y dispositivos virtuales: Android Studio incluye una gran variedad de emuladores de dispositivos Android que permiten a los desarrolladores probar sus aplicaciones con diferentes versiones de Android y configuraciones de hardware.

3.2. Lenguaje de programación: Java

Java [24] ha sido el lenguaje de programación tradicional y principal para el desarrollo de aplicaciones Android. Muchos desarrolladores están familiarizados con Java, y existe una amplia base de conocimientos y recursos disponibles para facilitar el desarrollo.

Java cuenta con un extenso ecosistema de bibliotecas y frameworks que pueden ser utilizados en proyectos Android. Esto incluye bibliotecas estándar de Java, así como frameworks específicos de Android que simplifican tareas comunes en el desarrollo de aplicaciones móviles.

3.3. API Spoonacular

Después de haber realizado la investigación de las API y bases de datos expuestas anteriormente, se realizó un listado definiendo qué características fundamentales tendría que tener la API para ser considerada ideal en el proyecto. El listado de las características es el siguiente:

- Permite la búsqueda de ingredientes y recetas, y que cada una de las recetas tenga información detallada respecto a ingredientes, valores nutricionales y preparación.
- Permite filtrar las recetas según el tipo de dieta (vegana, vegetariana, etc) y por alérgenos.
- Permite obtener imágenes de las diferentes recetas para que sea más intuitiva la visualización de las mismas por los usuarios.
- Permite su utilización de forma gratuita, ya que uno de los objetivos clave del proyecto es minimizar los costos. Por lo que cualquier API con prueba gratuita o con planes solamente de pago quedaría descartada.

- Y como característica adicional, se consideraría positivo que la API pueda generar dietas personalizadas a partir de diferentes restricciones.

Después de revisar las opciones y considerar nuestras necesidades, hemos seleccionado Spoonacular API como la principal fuente de datos. Esta API es gratuita y ofrece 150 peticiones al día, para obtener más se requiere un plan de pago.

Hay que destacar que con 150 peticiones no se pueden generar dietas de más de una semana, ya que la obtención de todas las recetas y sus correspondientes ingredientes consume muchas de éstas, pero para obtener una idea principal del funcionamiento de la aplicación sirve perfectamente.

El proceso de generar un menú para un día, que abarca tres comidas, cada vez que se solicita información sobre una receta o comida, obtener detalles sobre cada ingrediente presente en una receta y el acto de autogenerar un ingrediente a partir de una palabra específica, representan todas un coste de una petición en la API.

Por ejemplo, generar una dieta de 1 mes, requiere 30 peticiones para generar menús del día, más 90 peticiones para obtener la información relacionada para cada comida (30 días x 3 comidas al día), más 450 peticiones para obtener la información de cada ingrediente relacionado con una receta (teniendo en cuenta que una receta tiene una media de 5 ingredientes). Por lo tanto, el resultado sería de unas 570-600 peticiones, cosa que está muy por encima de las 150 peticiones diarias que proporciona el plan gratuito.

Además, contiene todas las características fundamentales establecidas anteriormente, incluso la opción de generar dietas personalizadas. Todas las recetas de la API buscan ser siempre lo más saludables posibles, por lo que estas dietas que genera automáticamente siempre van a ser lo más equilibradas posibles. Sin embargo,

es importante destacar que requiere atribución, lo que será considerado en el desarrollo de la aplicación.

La API contiene una página web donde se puede encontrar toda la documentación respecto a la base de datos y las diferentes peticiones que se pueden realizar [25]. Algunas de las más destacadas. y que se han usado en el proyecto son:

- Generar dietas según el tipo de dieta, las calorías objetivo y los alérgenos:

Con esta petición obtendremos una dieta, de un día o de una semana, según el atributo "timeFrame", compuesta por tres comidas, desayuno, comida y cena. Además, se puede indicar las calorías que se quiere ingerir en total al cabo del día con el atributo "targetCalories", el tipo de dieta mediante "diet" y los alérgenos mediante "exclude". Para realizar la petición se usa la siguiente url, con los datos que se quieran introducir:

`https://api.spoonacular.com/mealplanner/generate?timeFrame=${day/week}&targetCalories=${calories}&diet=${diet}&exclude=${ingredients}`

- Obtención de información detallada de una receta:

Con esta petición obtendremos todos los detalles acerca de la receta que se quiere obtener. Entre la información se puede encontrar los ingredientes, los pasos a seguir para realizar la receta, imágenes sobre la receta y sus valores nutricionales.

Para realizar la petición se usa la siguiente url, introduciendo el " id " de la receta que se quiere obtener:

`https://api.spoonacular.com/recipes/${id}/information`

Todas las solicitudes a la API devolverán los resultados en formato JSON. Esto es muy positivo de cara al proyecto, ya que facilita el tratamiento de los datos y su

integración sin problemas. El formato JSON es fácil de leer y manipular, lo que mejora la eficiencia y la usabilidad de la información obtenida de la API.

3.4. Herramientas de apoyo

A continuación, se van a explicar diferentes herramientas que se han utilizado durante el desarrollo como forma de apoyo.

3.4.1. Image Creator from Microsoft Designer

Esta herramienta [\[26\]](#) es capaz de generar imágenes basadas en descripciones textuales a través de inteligencia artificial. En nuestro proyecto ha sido utilizada para crear las imágenes de los avatares y el logo de la aplicación.

En cuanto a la propiedad intelectual de las imágenes generadas, estas no están protegidas expresamente por la ley de propiedad intelectual actual. Por ello, hemos podido utilizar las imágenes generadas en nuestro proyecto.

3.4.2. Git y Github

GitHub [\[27\]](#) es una plataforma de desarrollo colaborativo para alojar proyectos utilizando el sistema de control de versiones Git. En nuestro proyecto ha sido tremendamente útil para organizar el flujo de trabajo ya que nos ha permitido trabajar simultáneamente en el código sin tener que manejar manualmente las versiones.

3.4.3. DB Browser for SQLite

DB Browser for SQLite [\[28\]](#) es una herramienta de código abierto que permite visualizar y manipular bases de datos SQLite de manera gráfica e intuitiva. SQLite es un motor de base de datos ligero y autónomo que se utiliza en una amplia variedad de aplicaciones y sistemas, desde dispositivos móviles hasta aplicaciones de escritorio.

DB Browser for SQLite proporciona una interfaz gráfica que nos ha permitido visualizar, crear, editar y eliminar datos así como realizar consultas en la base de datos de nuestro proyecto, ya que facilita la visualización y manipulación de datos de una manera accesible y eficiente.

3.4.4. Google Fonts

Google Fonts [29] es una biblioteca en línea desarrollada por Google, en ella podemos encontrar diversas fuentes tipográficas de código abierto e iconos que se pueden utilizar de forma gratuita.

En el caso de nuestra aplicación, hemos obtenido de esta herramienta ciertos iconos que se muestran en los mensajes *pop-up* de la aplicación y son muy útiles para dar una representación visual a lo que se pretende transmitir. A continuación se muestran algunos ejemplos de iconos utilizados en nuestra aplicación:



Figura 3.1: Iconos de ejemplo

Capítulo 4: Casos de uso

En el marco de este proyecto, es esencial explorar y comprender la importancia de los casos de uso, una herramienta fundamental en el diseño y desarrollo de sistemas de software. Los casos de uso no solo son una representación gráfica de las interacciones entre usuarios y un sistema, sino que también desempeñan un papel crucial al delinear las funcionalidades clave y los escenarios de uso.

En su esencia, los casos de uso son una técnica que describe cómo un usuario interactúa con un sistema en particular para lograr un objetivo específico. Estos escenarios prácticos encapsulan las diversas interacciones entre los usuarios y la aplicación, brindando una visión clara de los requisitos funcionales y del comportamiento esperado del sistema.

En resumen, esta sección dedicada a los casos de uso desentraña su esencia y su papel esencial en el desarrollo de software, destacando cómo han sido empleados de manera efectiva en nuestro TFG para dar forma a una aplicación que responde a las demandas de sus usuarios.

4.1. Gestión de usuarios

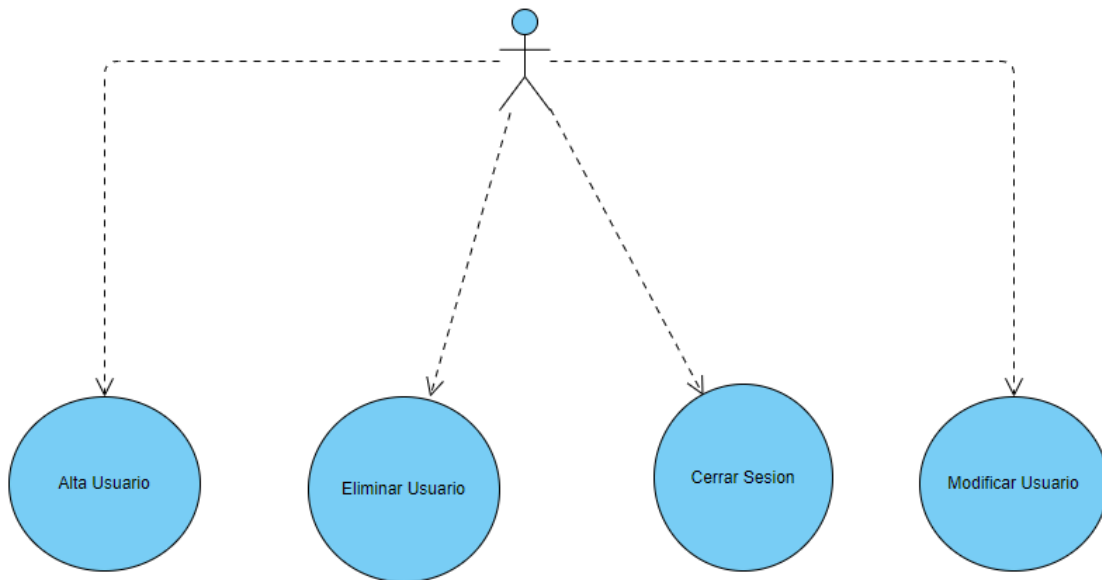


Figura 4.1: Diagrama UML de gestión de usuarios

4.1.1. Alta usuario

Actor Principal: Usuario

Objetivo: Permitir que un usuario se registre en la aplicación para acceder a funcionalidades personalizadas y guardar su información.

Precondiciones:

- El usuario ha descargado e instalado la aplicación.
- El usuario tiene acceso a una conexión a Internet.

Entradas: nombre de usuario, contraseña opcional, características personales sexo, peso y altura.

Salida: confirmación de registro.

Flujo Básico:

El usuario abre la aplicación en su dispositivo.

La aplicación presenta la opción de "+" en la pantalla de inicio.

El usuario selecciona la opción de "+".

El sistema presenta un formulario de registro solicitando la siguiente información:

- Nombre de usuario.
- Contraseña opcional.
- Selección de avatar(opcional).
- Características personales: sexo, peso y altura.

El usuario completa el formulario con la información requerida.

Si la validación es exitosa, el sistema crea una cuenta para el usuario y le proporciona un identificador único.

El usuario recibe una notificación de registro exitoso y es redirigido a la pantalla de inicio de sesión.

Postcondiciones:

- El usuario ha registrado una cuenta en la aplicación.
- El usuario puede acceder a funcionalidades personalizadas y guardar su información.
- La información del usuario se almacena de manera segura en la base de datos de la aplicación.

4.1.2. Modificación de perfil de usuario

Actor Principal: Usuario registrado

Objetivo: Permitir que un usuario modifique la información de su perfil en la aplicación.

Precondiciones:

- El usuario ha iniciado sesión en la aplicación.
- El usuario tiene acceso a una conexión a Internet.

Entradas: nuevo nombre que proporcione el usuario, nueva contraseña, nuevo avatar a seleccionar, nuevo peso o nueva altura.

Salida: actualización del perfil correctamente.

Flujo Básico:

El usuario accede a la sección de "Settings" dentro de la aplicación.

La aplicación presenta la opción de "Modify Profile".

El usuario selecciona la opción de "Modify Profile".

El sistema muestra un formulario con la información del perfil del usuario actual, que puede incluir:

- Nombre completo.
- Contraseña Opcional
- Avatar a cambiar
- Peso
- Altura
- Edad

El usuario modifica la información necesaria en el formulario.

El sistema valida la información modificada.

Si la validación es exitosa, el sistema actualiza la información del perfil con los nuevos datos.

El usuario recibe una notificación de que la modificación del perfil fue exitosa y puede ver la información actualizada en la sección de perfil.

Postcondiciones:

- La información del perfil del usuario se ha actualizado con éxito.
- El usuario puede acceder a las funcionalidades de la aplicación con la información actualizada.
- Los cambios realizados se reflejan en la base de datos de la aplicación.

4.1.3. Eliminación de perfil de usuario

Actor Principal: Usuario registrado

Objetivo: Permitir que un usuario elimine permanentemente su perfil y datos asociados en la aplicación.

Precondiciones:

- El usuario ha iniciado sesión en la aplicación.
- El usuario tiene acceso a una conexión a Internet.

Entrada: mensaje de confirmación de eliminación del usuario.

Salida: eliminación del perfil correctamente.

Flujo Básico:

El usuario accede a la sección de "Perfiles" dentro de la aplicación.

La aplicación presenta el icono de la papelera. El usuario selecciona la opción "del icono de papelera".

El sistema muestra una confirmación para asegurarse de que el usuario desea proceder con la eliminación del perfil advirtiéndole de la pérdida de dietas configuradas.

El usuario confirma la eliminación del perfil.

El usuario recibe una notificación confirmando la eliminación exitosa y es redirigido a la pantalla de inicio o a una página de cierre de sesión.

Postcondiciones:

- El perfil del usuario y todos los datos asociados se eliminan permanentemente.
- El usuario ya no tiene acceso a la aplicación con esa cuenta.
- Cualquier información relacionada con el usuario se elimina de la base de datos de la aplicación.

4.1.4. Cierre de sesión en la aplicación

Actor Principal: Usuario registrado

Objetivo: Permitir que el usuario cierre sesión de manera segura en la aplicación.

Precondiciones:

- El usuario ha iniciado sesión en la aplicación.

Entrada: mensaje de confirmación para cerrar la sesión.

Salidas: redirección a la pantalla principal de usuarios, eliminación de la sesión activa.

Flujo Básico:

El usuario accede a la sección de "Perfil" dentro de la aplicación.

La aplicación presenta la opción de "Logout".

El usuario selecciona la opción "Logout".

El sistema confirma con el usuario si realmente desea cerrar sesión para garantizar que la acción es intencional.

El usuario confirma el cierre de sesión.

El sistema cierra la sesión del usuario y lo redirige a la pantalla de inicio o a una pantalla de inicio de sesión.

Postcondiciones:

- La sesión del usuario se cierra de manera segura.
- El usuario ya no tiene acceso a las funcionalidades protegidas por inicio de sesión.
- Si el usuario decide cerrar la aplicación, deberá volver a iniciar sesión en su próxima visita.

4.2. Gestión de Dietas

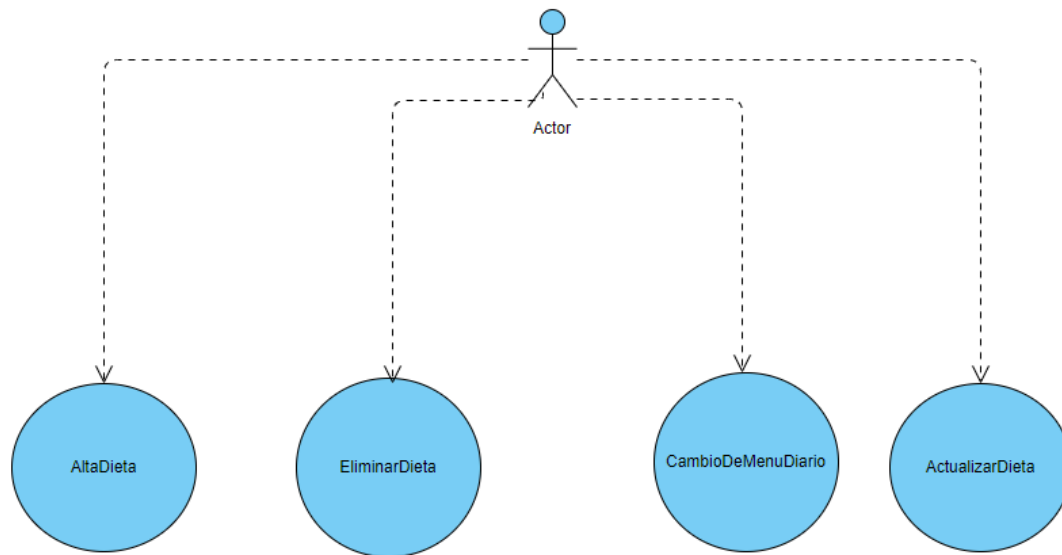


Figura 4.2: Diagrama UML de gestión de dietas

4.2.1. Creación de una dieta

Actor Principal: Usuario

Objetivo: El usuario utiliza la aplicación para planificar y seguir una dieta personalizada.

Precondiciones:

- El usuario ha iniciado sesión en la aplicación.
- El usuario ha proporcionado información personal como peso, altura y objetivos.

Entradas: introducción de datos del usuario, selección del tipo de dieta correspondiente, introducción de alimentos a evitar en la dieta de usuarios.

Salida: dieta creada correctamente

Flujo Básico:

El usuario ingresa a la aplicación y accede a la sección de "Food Calendar".

El usuario selecciona la opción "Create New Diet".

El usuario proporciona información adicional según sea necesario, edad, peso, sexo, altura y actividad física a la semana.

El sistema solicita al usuario que ingrese preferencias adicionales, como restricciones alimenticias, preferencias de alimentos, tipos de dietas etc.

El sistema genera una dieta personalizada basada en la información proporcionada por el usuario.

El usuario revisa la dieta propuesta y tiene la opción de realizar ajustes según sus preferencias.

Una vez satisfecho, el usuario confirma la dieta y la guarda en su perfil.

Postcondiciones:

- La dieta personalizada se guarda en el perfil del usuario.
- El usuario puede acceder a la dieta planificada en cualquier momento.
- El sistema puede proporcionar recordatorios y seguimiento del progreso según la dieta planificada.

4.2.2. Eliminación de dieta

Actor Principal: Usuario registrado

Objetivo: Permitir que el usuario elimine una dieta específica que ha sido creada o seleccionada previamente en la aplicación.

Precondiciones:

- El usuario ha iniciado sesión en la aplicación.
- El usuario tiene al menos una dieta creada o seleccionada.

Entrada: mensaje de confirmación de eliminación de dieta.

Salida: dieta eliminada correctamente.

Flujo Básico:

El usuario accede a la sección de "Settings" en la aplicación.

El usuario selecciona el botón "Delete my Diet".

El sistema muestra una confirmación para asegurarse de que el usuario desea proceder con la eliminación de la dieta.

El usuario confirma la eliminación de la dieta.

El sistema elimina la dieta seleccionada de la lista del usuario.

Postcondiciones:

- La dieta seleccionada se elimina de la lista del usuario.
- Cualquier progreso o datos asociados a la dieta se eliminan o archivan según la implementación.
- La aplicación actualiza la interfaz para reflejar la eliminación de la dieta.

4.2.3. Cambio de menú diario

Actor Principal: Usuario

Objetivo: El usuario utiliza la aplicación para cambiar el menú que le ha sido asignado.

Precondiciones:

- El usuario ha iniciado sesión en la aplicación.
- El usuario ha proporcionado información personal como peso, altura y objetivos.
- El usuario ha creado una dieta en nuestra aplicación.

Entradas: introducción de datos del usuario, selección del tipo de dieta correspondiente, introducción de alimentos a evitar en la dieta de usuarios.

Salida: nuevo menú generado correctamente.

Flujo Básico:

El usuario ingresa a la aplicación y accede a la sección de "Food Calendar". O bien, accede a la sección "Summary" y pulsa en el enlace "Click here to show it".

El usuario selecciona la opción "New".

El sistema genera un nuevo menú personalizado basado en la información proporcionada por el usuario.

Una vez satisfecho, el nuevo menú se guarda en su perfil.

Postcondiciones:

- El nuevo menú se guarda en el perfil del usuario.
- El usuario puede acceder a su nuevo menú en cualquier momento.

4.3. Seguimiento de peso

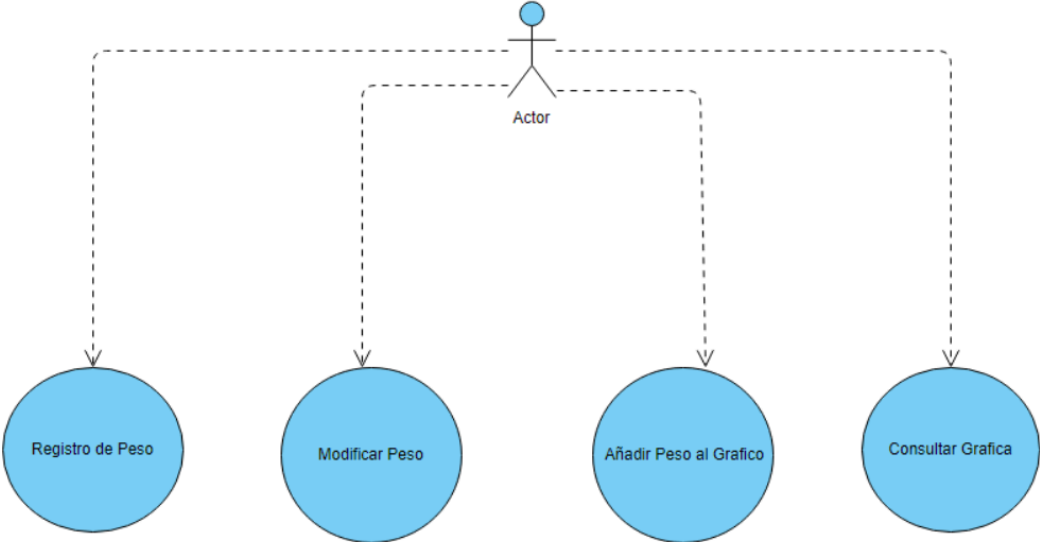


Figura 4.3: Diagrama UML de seguimiento de peso

4.3.1. Registro de peso de un usuario

Actor Principal: Usuario registrado

Objetivo: Permitir que el usuario añada su peso actual en el calendario

Precondiciones:

- El usuario ha iniciado sesión en la aplicación.
- El usuario tiene al menos una dieta creada o seleccionada.

Flujo Básico:

El usuario accede a la sección de "Weight progress" en la aplicación.

El usuario selecciona una fecha.

El usuario registra el peso actual en kilogramos.

El sistema muestra una confirmación para asegurarse de que el usuario está seguro de haber puesto su peso adecuadamente.

El usuario confirma su peso.

El sistema añade su peso a la base de datos.

Postcondiciones:

- El peso queda registrado en el sistema.
- Cualquier progreso o datos asociados al peso se archivan según la implementación.
- La aplicación actualiza la interfaz para reflejar la inserción de peso correspondiente.

4.3.2. Modificación de peso de un usuario

Actor Principal: Usuario registrado

Objetivo: Permitir que el usuario modifique su peso en el calendario a medida que avanza con su dieta correspondiente.

Precondiciones:

- El usuario ha iniciado sesión en la aplicación.
- El usuario tiene al menos una dieta creada o seleccionada.

Flujo Básico:

El usuario accede a la sección de "Weight progress" en la aplicación.

El usuario selecciona una fecha

El usuario modifica su peso.

El sistema muestra una confirmación para asegurarse de que el usuario desea proceder con la modificación de su peso.

El usuario confirma la modificación de su peso.

El sistema cambia su peso actual por el registrado anteriormente.

Postcondiciones:

- El nuevo peso queda registrado en el sistema.
- Cualquier progreso o datos asociados al peso se archivan o se pueden modificar según la implementación.
- La aplicación actualiza la interfaz para reflejar la nueva modificación del peso añadida por el usuario

4.3.3. Consultar gráfica de peso

Actor Principal: Usuario registrado

Objetivo: Permitir que el usuario vea su avance de peso en relación a su dieta.

Precondiciones:

- El usuario ha iniciado sesión en la aplicación.
- El usuario tiene al menos una dieta creada o seleccionada.
- El usuario ha registrado su peso en el día correspondiente en la sección "Food Calendar".

Flujo Básico:

El usuario accede a la sección de "Weight progress" en la aplicación.

El usuario selecciona el tiempo con el que quiere ver la gráfica. P.e; una semana, un mes o el histórico.

El usuario tiene la opción de ver dos líneas; su peso con la línea de color azul y su peso estimado con la línea de color verde.

Postcondiciones:

- El peso queda registrado en el sistema.
- Cualquier progreso o datos asociados al peso se archivan según la implementación.
- La aplicación actualiza la interfaz para reflejar la inserción de peso correspondiente.

4.3.4. Añadir peso al gráfico

Actor Principal: Usuario registrado

Objetivo: Permitir que el usuario añada su peso actual en el calendario

Precondiciones:

- El usuario ha iniciado sesión en la aplicación.
- El usuario tiene al menos una dieta creada o seleccionada.

Flujo Básico:

El usuario accede a la sección de "Weight progress" en la aplicación.

El usuario selecciona una fecha.

El usuario registra el peso actual en kilogramos.

El sistema muestra una confirmación para asegurarse de que el usuario está seguro de haber puesto su peso adecuadamente.

El usuario confirma su peso.

El sistema añade su peso a la base de datos.

Postcondiciones:

- El peso queda registrado en el sistema.
- Cualquier progreso o datos asociados al peso se archivan según la implementación.
- La aplicación actualiza la interfaz para reflejar la inserción de peso correspondiente.

4.4. Gestión de la lista de la compra

Actor Principal: Usuario registrado

Objetivo: Permitir que el usuario vea los ingredientes de su dieta para ver cuales tiene que comprar para preparar el menú correspondiente.

Precondiciones:

- El usuario ha iniciado sesión en la aplicación.
- El usuario tiene al menos una dieta creada o seleccionada.

Flujo Básico:

El usuario accede a la sección de "Shopping List" en la aplicación.

El usuario puede consultar los ingredientes que necesita en función del tipo de comida(Desayuno, Comida y Cena) y en función del tiempo (3, 5 o 7 días). Aparte puede ver todos los ingredientes o seleccionar en función si no los necesita comprar o tiene que comprarlos.

Postcondiciones:

- Los ingredientes marcados quedan registrados en el sistema.

5.1. Administración de perfiles

La primera vez que el usuario ingrese a la aplicación, se mostrará la pantalla principal de perfiles (ver figura 5.2). En ella se solicitará al usuario crear un perfil para empezar a disfrutar de la aplicación.

Los usuarios se muestran en una tabla en la pantalla principal de perfiles, en el cuadro de cada perfil se muestra el avatar, nombre y un icono de candado si este perfil tiene contraseña.

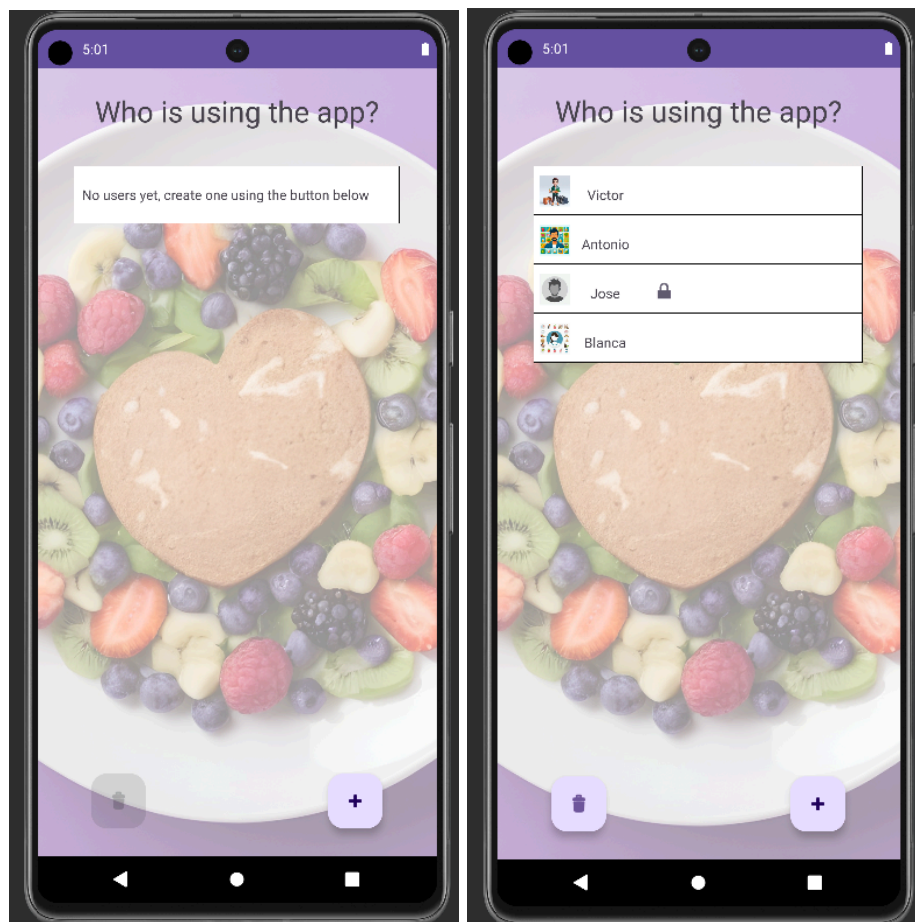


Figura 5.2: Pantalla acceso a usuarios

Creación de un perfil

En la parte inferior del menú principal de perfiles, observamos dos botones, si pulsamos sobre el símbolo "+", este nos llevará a la pantalla de creación de un perfil (ver figura 5.3), en ella el usuario podrá introducir su nombre, una contraseña opcionalmente y elegir un avatar para su perfil, que podrá elegir entre subir una imagen desde su almacenamiento o uno de los predeterminados de la aplicación.

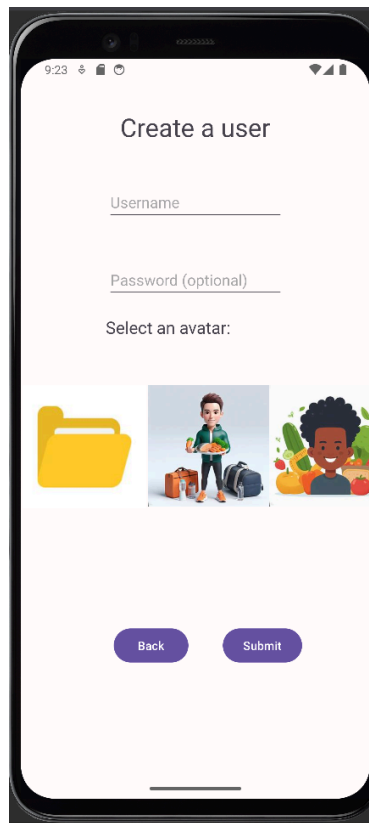


Figura 5.3: Pantalla crear un usuario

Borrado de un perfil

Si en este caso pulsamos en el botón con el cubo de basura, este hará que sobre los perfiles aparezca un icono para eliminar el usuario (ver figura 5.4). En el caso de los usuarios con contraseña, se le pedirá al usuario que la introduzca.

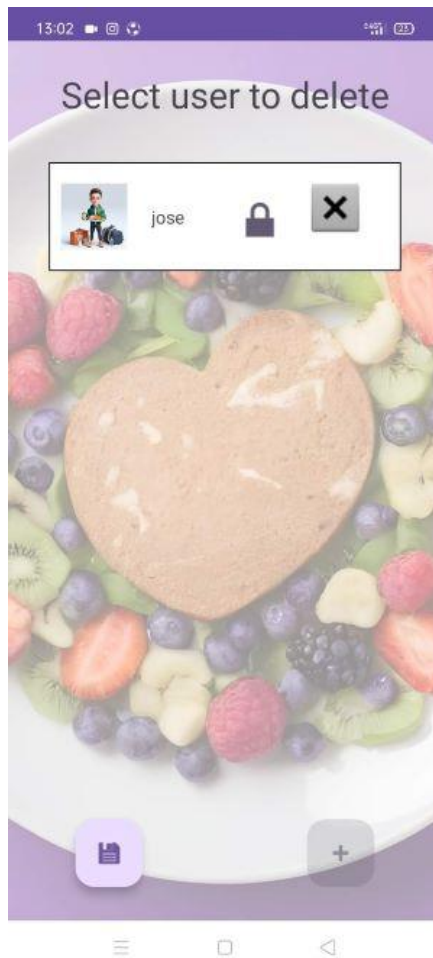


Figura 5.4: Pantalla borrado de un perfil

5.2. Pantalla principal y navegación

Una vez iniciada la sesión, el usuario accederá a la pantalla principal (ver figura 5.5) de la aplicación, en ella se muestra un consejo del día en función a tu tipo de dieta elegido y accesos rápidos, como enlaces a la eliminación de una dieta y la consulta del menú.



Figura 5.5: Pantalla principal de la aplicación

Deslizando desde la izquierda se despliega este menú de navegación en el que se le presentan al usuario todas las opciones que tiene dentro de la aplicación (ver figura 5.6).

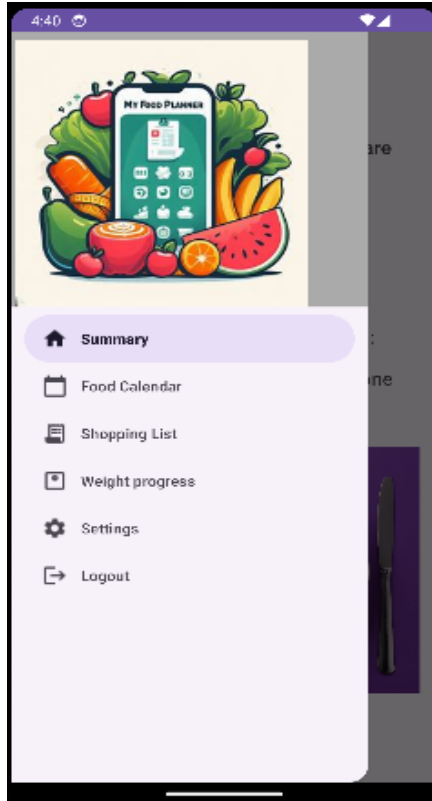


Figura 5.6: Menú desplegable de navegación

5.3. Generación de una dieta

A continuación, se va a explicar cómo se genera una dieta en nuestra aplicación y los pasos a seguir para poder ver los diferentes menús generados, además de las diferentes funcionalidades relacionadas con la misma.

Introducción de información personal

Cuando un usuario haya creado un perfil, se le dirigirá directamente a la pantalla mostrada en la figura 5.6 donde podrá introducir su información fisiológica personal, como edad, altura, sexo y por último actividad física semanal. Si no es la primera vez que visita esta pantalla, los datos introducidos previamente se mostrarán en ella. Estos datos se almacenarán con el fin de crear una dieta acorde a las necesidades fisiológicas del usuario.

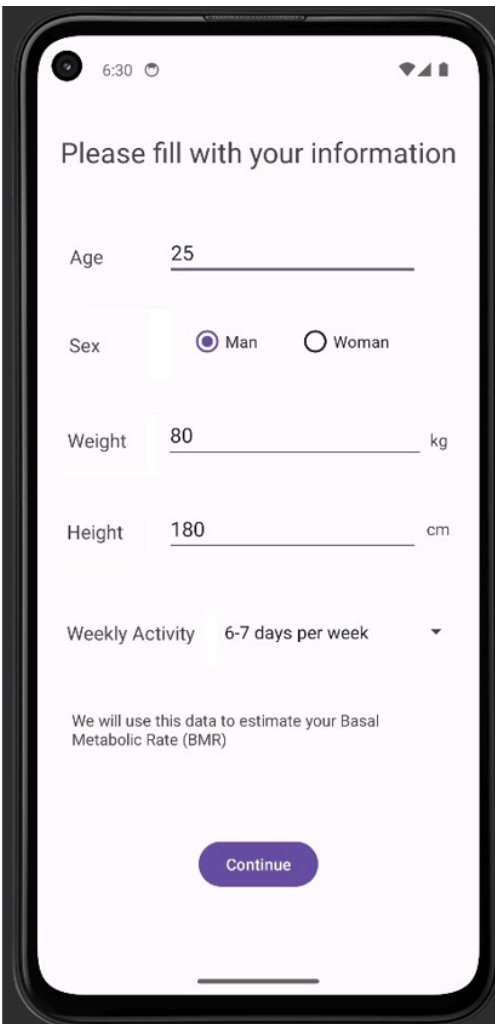


Figura 5.7: Pantalla de introducción de datos

En la pantalla de la figura 5.7, el usuario podrá introducir las calorías a consumir diariamente, o por otro lado elegir que se calculen automáticamente según el objetivo de su dieta, en función de si quiere perder o mantener el peso, o ganar masa muscular.

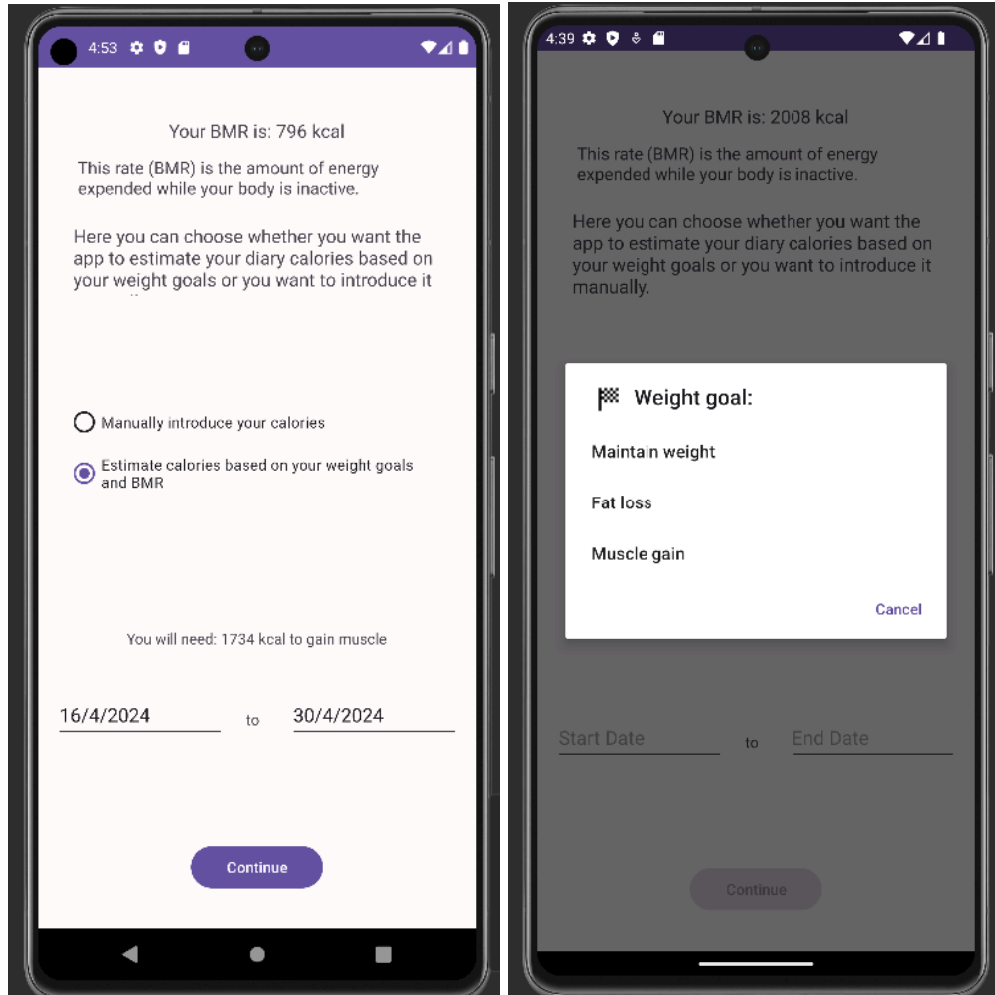


Figura 5.8: Pantalla de opciones de creación de dieta

En la pantalla de la figura 5.9, el usuario podrá introducir el tipo de dieta que quiere, e incluir mediante un autocompletar, las diferentes comidas o ingredientes que no le gustan o no quiere ingerir.

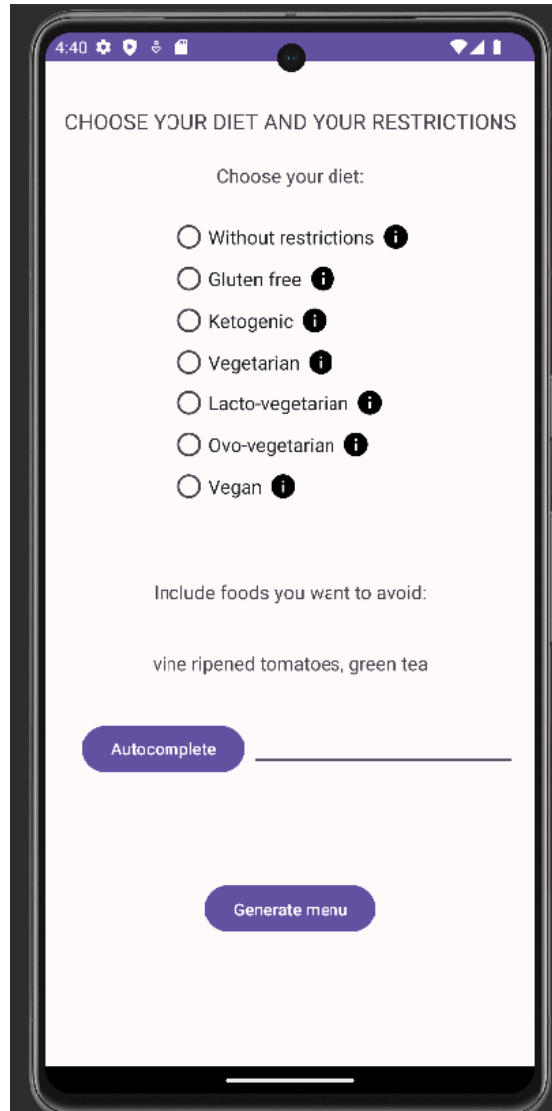


Figura 5.9: Pantalla de introducción de tipo de dieta

5.4. Visualización del menú diario

Si pulsamos en el apartado “Food Calendar” de la navegación lateral (ver figura 5.6), este nos llevará a la pantalla de la figura 5.10 en ella el usuario podrá pulsar en cualquier día del calendario y ver las comidas que tiene asignadas para ese día.

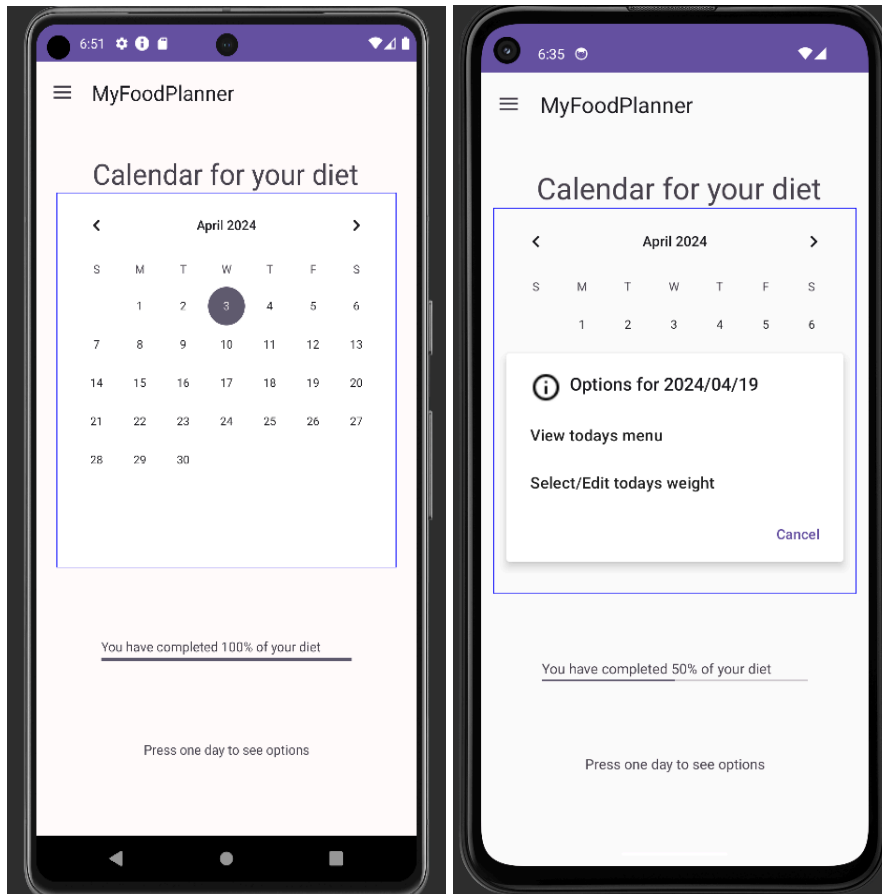


Figura 5.10: Calendario de dieta

Una vez seleccionado el día al que se quiere acceder, como se puede ver en la figura 5.11, se van a mostrar las diferentes comidas que tiene para ese día, indicando el título, sus calorías y una foto de cada una de ellas.

Además, en esta vista, se puede realizar la generación de un nuevo menú, mediante el botón "new", en caso de que el usuario no esté satisfecho con el que tiene actualmente.

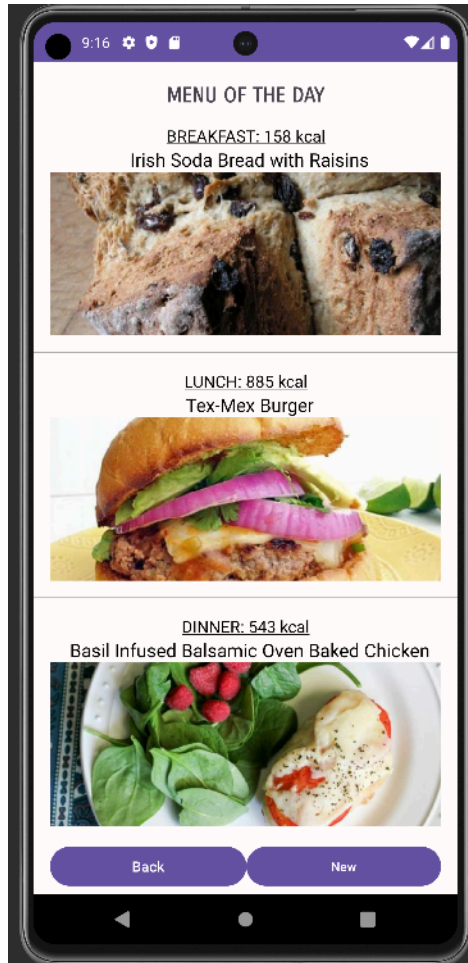


Figura 5.11: Visualización de un menú diario

Si el usuario pulsa en alguna de las tres comidas que tiene el menú, se le mostrará información relacionada con esa comida, como se puede ver en la figura 5.12, indicando el título, una imagen, una pequeña descripción, y la lista de ingredientes necesaria para realizar la receta y sus respectivos pasos a seguir para preparar dicho plato, pulsando en los botones de la parte inferior.

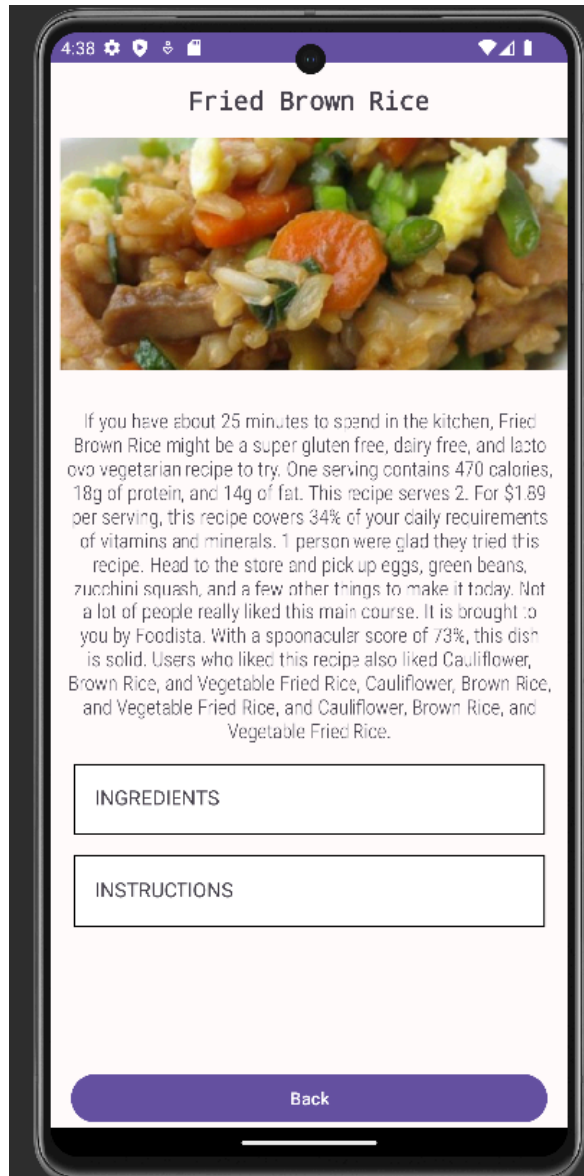


Figura 5.12: Visualización de un plato individual

5.5. Lista de la compra

Pulsando en la sección "Shopping List" de la barra de navegación de la pantalla principal (ver figura 5.6), este nos llevará a la pantalla de la figura 5.13, donde se pueden apreciar todos los ingredientes correspondientes a nuestra dieta. Los ingredientes aparecen en orden alfabético y divididos en tres secciones, según si pertenecen al desayuno, a la comida o a la cena.

Todos los ingredientes se pueden marcar o desmarcar, para indicar de forma sencilla y rápida, que ya se tiene ese ingrediente.

Además, la interfaz tiene dos filtros muy útiles para usar de una manera más sencilla la lista de ingredientes. Con el primer filtro, se puede indicar si se quiere mostrar los ingredientes marcados, desmarcados o todos los ingredientes. Y el segundo filtro, permite seleccionar la cantidad de días para mostrar los ingredientes. Puedes elegir entre 3, 5 o 7 días para ver los ingredientes correspondientes a ese período de tiempo. Esto te ayudará a planificar tus compras y comidas de manera más eficiente según tus necesidades y preferencias.

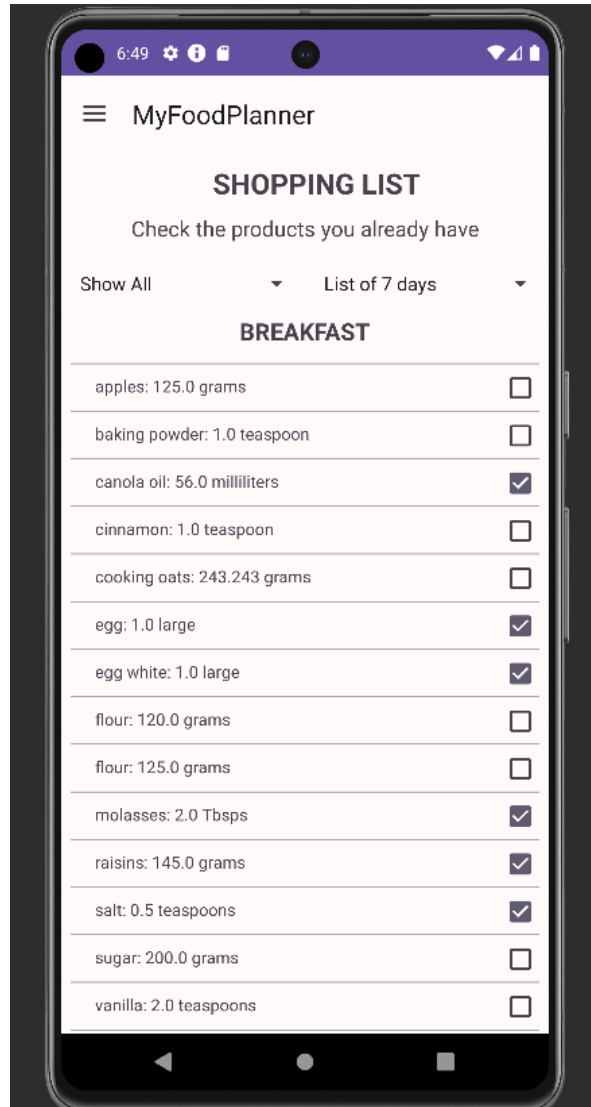


Figura 5.13: Pantalla de selección de ingredientes

5.6. Seguimiento de peso

Si pulsamos en el apartado “Weight Monitoring” de la navegación lateral, este nos llevará a la pantalla de la figura 5.14 donde el usuario podrá visualizar a través de un gráfico de puntos (ver figura 5.14), su progresión de peso. También se le permite al usuario elegir el periodo de tiempo para el que quiere mostrar los pesos, pudiendo elegir entre los últimos 7 días, el último mes o el histórico de pesos.

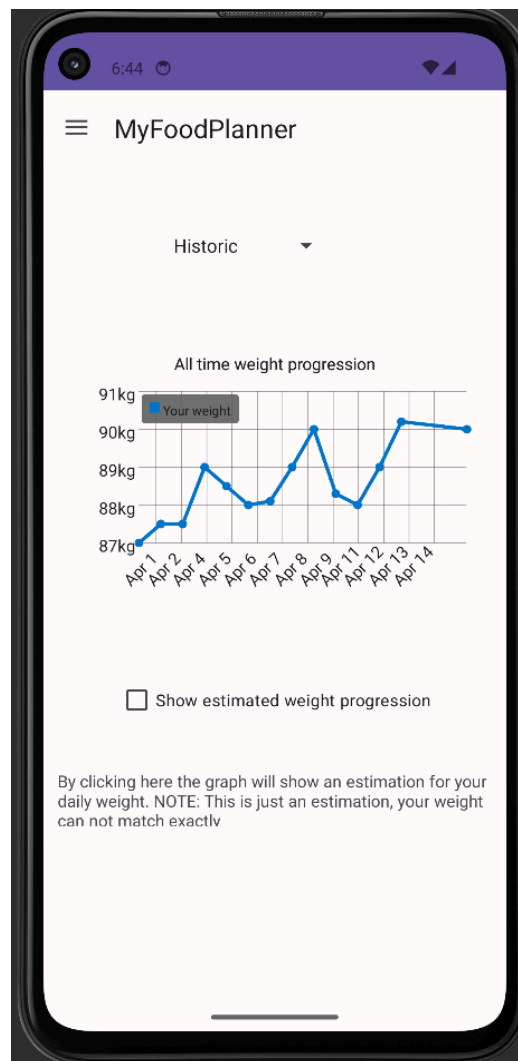


Figura 5.14: Pantalla principal de visualización de progresión

5.7. Evolución óptima de peso

El usuario tiene la posibilidad de mostrar una estimación de la evolución de peso que debería seguir. Esta se calcula según el objetivo que tenga la dieta del perfil activo y se muestra como una serie de puntos de color verde en la gráfica de peso. A continuación se muestra en la figura 5.15:

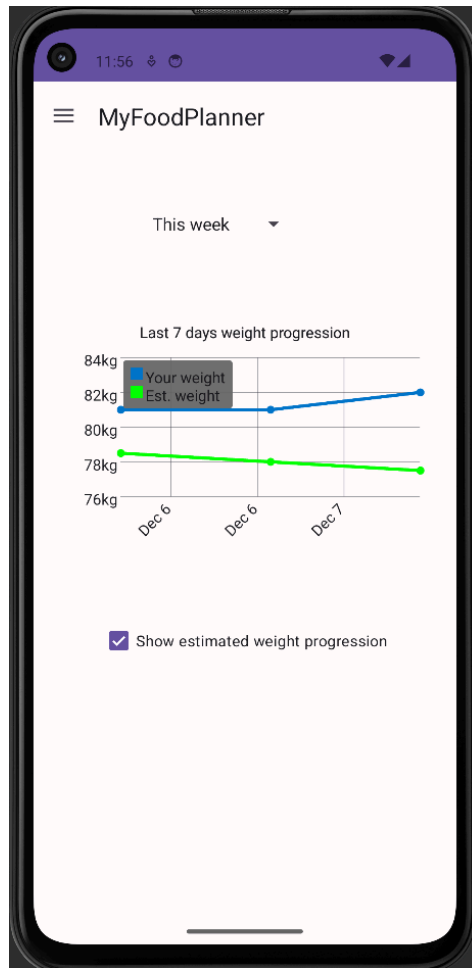


Figura 5.15: Visualización de serie de peso óptimo

Capítulo 6: Detalles de implementación

En este apartado se van a explicar cómo el modelo de datos implementado o diferente código para realizar las peticiones a la Application Programming Interface.

6.1. Modelo de datos

En este apartado vamos a explicar cómo está estructurada la base de datos de la aplicación. Esta consta de una serie de tablas que nos permiten almacenar todos los datos sobre usuarios, dietas y pesos.

En primer lugar y como epicentro del almacenamiento de datos, encontramos la tabla *t_usuarios*, que almacena los datos básicos de los usuarios de la aplicación. En ella encontramos columnas como nombre de usuario, identificador único, que es la primary key de la tabla, avatar, contraseña, edad, peso, altura, sexo, los alimentos no deseados y la actividad semanal del usuario. En el caso de la contraseña, decidimos no hacerla obligatoria ni cifrarla, ya que la aplicación se ha diseñado para uso local y no se considera necesario seguridad extra. La tabla se muestra a continuación:

name	id (PK)	avatar	password	age	height	weight	gender	activity	excluded
------	---------	--------	----------	-----	--------	--------	--------	----------	----------

Tabla 6.1: Tabla *t_usuarios*

En segundo lugar encontramos la tabla *t_dietas*, esta tabla nos permite almacenar las diferentes dietas de los usuarios de la aplicación. Dispone de 6 columnas; un identificador único primary key, el tipo de dieta, el número de kcal diarias a consumir, el identificador del usuario al que le pertenece la dieta y las fechas de inicio y final de la dieta. El campo *userID* es una foreign key (FK), referida al campo *id* de la tabla *t_usuarios*. La tabla se muestra a continuación:

id (PK)	type	kCalories	userID (FK)	startDate	endDate
---------	------	-----------	-------------	-----------	---------

Tabla 6.2: Tabla *t_dietas*

En tercer lugar explicamos la tabla *t_dietmeals*, con esta tabla almacenamos las comidas diarias para todos los días de una dieta, de esta manera podemos almacenar y acceder de manera sencilla a las comidas de cada día. En esta tabla también almacenamos las cantidades de macronutrientes que se han estimado para ese día de dieta del usuario. Como primary key encontramos la combinación única de id y día de dieta, de esta manera solo existe una fila para cada día de una dieta en concreto. En cuanto a las foreign keys, encontramos breakfastID, lunchID y dinnerID, estos tres campos representan el id de cada una de las tres comidas asignadas para un día de esa dieta. A continuación se muestra la tabla:

id	day	fat	ch	prot	breakfastID (FK)	lunchID (FK)	dinnerID (FK)	dietType
----	-----	-----	----	------	------------------	--------------	---------------	----------

Tabla 6.3: Tabla *t_dietmeals*

En cuarto lugar encontramos la tabla *t_meal*, esta tabla almacena toda la información de una comida: id único y PK, nombre, ingredientes, instrucciones, cantidad de grasa, proteínas y carbohidratos, receta, tipo de imagen y calorías. La tabla se muestra a continuación:

id (PK)	name	ing	instr	fatQ	protQ	chQ	recipe	typeIMG	cal
---------	------	-----	-------	------	-------	-----	--------	---------	-----

Tabla 6.4: Tabla *t_meal*

En quinto lugar, explicamos la tabla *t_ingredientsmeals*, que almacena la información sobre los ingredientes de una comida. Esta tabla tiene una columna para identificar la dieta a la que pertenece el ingrediente, un identificador único y PK, cantidad y otro identificador de la receta a la que pertenece. También hay una columna que indica si el usuario ha marcado este ingrediente en el apartado de lista de la compra, de esta manera almacenamos esta información. La tabla a continuación representa esta información:

id (PK)	idRecipe	name	amount	unit	marked
---------	----------	------	--------	------	--------

Tabla 6.5: Tabla *t_ingredientsmeals*

En sexto lugar, encontramos la tabla de seguimiento de peso *t_weightmonitoring*, esta tabla almacena el peso de un usuario para un día determinado. Su primary key es el conjunto de userID y día, de esta manera cada usuario tiene un único peso por cada día. La tabla se muestra a continuación:

userID	weight	day
--------	--------	-----

Tabla 6.6: Tabla *t_weightmonitoring*

Por último, la tabla *t_advices*, que almacena los consejos que se muestran al usuario. Para ello almacena un identificador único (PK), un nombre y el tipo de dieta al que aconseja. A continuación se muestra la tabla:

id (PK)	name	dietType
---------	------	----------

Tabla 6.7: Tabla *t_advices*

Modelo entidad-relación

El modelo entidad-relación representa las entidades (objetos o conceptos del mundo real) y sus relaciones en un sistema. En el contexto de una aplicación Android, el modelo entidad-relación ayuda a visualizar cómo se estructuran los datos dentro de la base de datos de la aplicación. Al identificar las entidades principales y sus interacciones, se puede comprender mejor la lógica subyacente de la aplicación y diseñar una base de datos eficiente y coherente. A continuación se muestra en la figura 5.16 el modelo entidad-relación para nuestra aplicación:

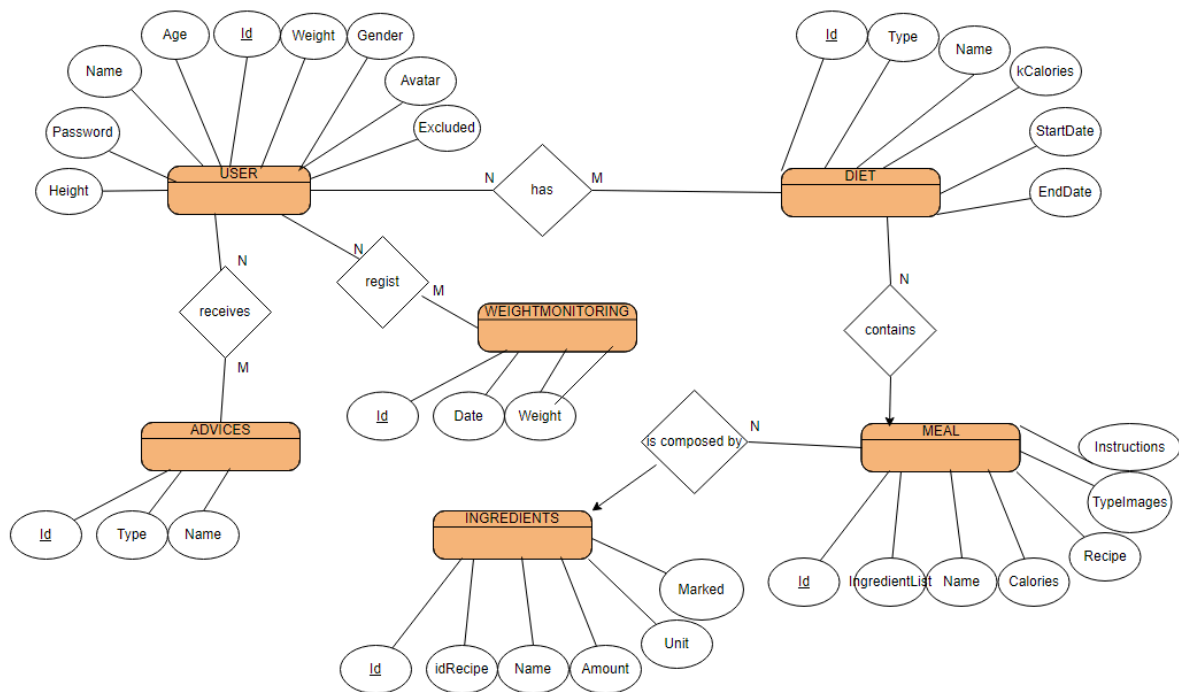


Figura 6.1: Modelo entidad-relación

En él se puede observar cómo se relacionan las entidades, en el caso de un usuario, este registra su peso diario, recibe consejos que se almacenan en la base de datos y

tiene una dieta única. Esta dieta a su vez, contiene una serie de comidas que se distribuyen por días, y estas comidas tienen también unos ingredientes que se necesitan para su preparación.

6.2. Detalles de implementación

Petición a la Application Programming Interface

Una vez obtenidos todos los datos relacionados con la dieta y el usuario, se realizará la llamada a la Application Programming Interface como se muestra en la figura 5.17. En esta petición se incluye un JSONObject, donde se guardan los datos obtenidos para posteriormente se realizar diferentes funciones con ellos, un ErrorListener, para controlar excepciones en caso de que se produzcan errores durante la petición, las calorías de la dieta del usuario, el tipo de dieta y por último, el listado de comidas que no quiere el usuario.

```
public void mealPlanner(Response.Listener<JSONObject> listener,
    Response.ErrorListener errorListener,
    int calories,
    String diet,
    String foods) {
    String url = BASE_URL + "/mealplanner/generate?timeFrame=day&targetCalories="
        + calories + "&diet=" + diet + "&exclude=" + foods + "&apiKey=" + API_KEY;
    JSONObjectRequest request = new JSONObjectRequest(Request.Method.GET, url, jsonRequest: null, listener, errorListener);
    requestQueue.add(request);
}
```

Figura 6.2: Petición a la API para generar una dieta

Lo que devuelve la petición es un listado de tres recetas, una para cada comida del día, y los respectivos nutrientes del menú del día generado, como se puede ver en la figura 5.18.

```

{
  "meals": [
    {
      "id": 657359,
      "imageType": "jpg",
      "title": "Pumpkin Pie Smoothie",
      "readyInMinutes": 5,
      "servings": 2,
      "sourceUrl": "https://spoonacular.com/pumpkin-pie-smoothie-657359"
    },
    {
      "id": 1697583,
      "imageType": "jpg",
      "title": "Crispy Ravioli Formaggi",
      "readyInMinutes": 30,
      "servings": 2,
      "sourceUrl": "https://spoonacular.com/crispy-ravioli-formaggi-1697583"
    },
    {
      "id": 715438,
      "imageType": "jpg",
      "title": "Mexican Casserole",
      "readyInMinutes": 65,
      "servings": 6,
      "sourceUrl": "https://spoonacular.com/mexican-casserole-715438"
    }
  ],
  "nutrients": {
    "calories": 2000.03,
    "protein": 90.84,
    "fat": 120.98,
    "carbohydrates": 141.94
  }
}

```

Figura 6.3: Resultado de generar una dieta un menú de un día

Después de generar la dieta y sus respectivas recetas, se realizará la llamada a la Application Programming Interface como se muestra en la figura 5.19, para obtener toda la información respecto a dichas recetas. En esta petición se incluye un JSONObject, donde se guardan los datos obtenidos para posteriormente se realizar diferentes funciones con ellos y un ErrorListener, para controlar excepciones en caso de que se produzcan errores durante la petición.

El resultado que devuelve la petición es toda la información respecto a la dieta, su nombre, imagen, listado de todos sus ingredientes, instrucciones para realizar la receta, información nutricional y muchos datos más.

```

public void informationRecipe(int recipeId, Response.Listener<JSONObject> listener, Response.ErrorListener errorListener) {
    String url = BASE_URL + "/recipes/" + recipeId + "/information?apiKey=" + API_KEY;
    JSONObjectRequest request = new JSONObjectRequest(Request.Method.GET, url, null, new Response.Listener<JSONObject>() {
        @Override
        public void onResponse(JSONObject response) {
            try {
                // Extraer las instrucciones y el resumen
                String title = response.getString("title");
                String typeImage = response.getString("image");
                String instructions = response.getString("instructions");
                // Extraer la lista de ingredientes
                JSONArray ingredientsArray = response.getJSONArray("extendedIngredients");
            }
        }
    });
}

```

Figura 6.4: Petición a la API para obtener información de una receta

Para incluir los ingredientes no deseados por el usuario, se realizará la llamada a la Application Programming Interface como se muestra en la figura 5.20, donde se incluirá una palabra o ingrediente escrito por el usuario, y devolverá un listado con ingredientes que coincidan con el proporcionado por el usuario, como se puede ver en la figura 5.21. En esta petición se incluye un JSONArray, donde se guardan los datos obtenidos para posteriormente se realizar diferentes funciones con ellos, un ErrorListener, para controlar excepciones en caso de que se produzcan errores durante la petición y el ingrediente, el cual se quiere buscar en la base de datos de la API.

```

public void autocompleteIngredients(Response.Listener<JSONArray> listener,
    Response.ErrorListener errorListener, String ingredient) {
    String url = BASE_URL + "/food/ingredients/autocomplete?metaInformation=false&number=4&query="
        + ingredient + "&apiKey=" + API_KEY;
    JSONArrayRequest request = new JSONArrayRequest(url, listener, errorListener);
    requestQueue.add(request);
}

```

Figura 6.5: Petición a la API para autocompletar ingredientes

```
[
  {
    "name": "egg",
    "image": "egg.png"
  },
  {
    "name": "eggnog",
    "image": "eggnog.png"
  },
  {
    "name": "egg yolk",
    "image": "egg-yolk.jpg"
  },
  {
    "name": "eggplant",
    "image": "eggplant.png"
  }
]
```

Figura 6.6: Resultado de petición autocompletar un ingrediente

Capítulo 7: Test en usuarios

En este capítulo, se detalla el proceso de evaluación de la aplicación mediante pruebas realizadas con usuarios. Se describirán los objetivos del testeo y la metodología empleada. Además, se presentarán los resultados obtenidos, que permitirán entender la efectividad y usabilidad de la aplicación desde la perspectiva de los usuarios finales. Este análisis es fundamental para identificar áreas de mejora y asegurar que la aplicación cumple con las expectativas y necesidades del público objetivo.

7.1. Generación de la APK

Para garantizar la funcionalidad y usabilidad de nuestra aplicación móvil, hemos generado una APK (Android Package Kit) que nos ha permitido probar la aplicación en diferentes dispositivos. Este paso ha sido muy importante para identificar y corregir diversos errores, principalmente, relacionados con la interfaz de usuario, optimizar el rendimiento y mejorar la experiencia del usuario.

Durante todo el proceso de desarrollo se han generado un total de 10 versiones de la APK hasta obtener la versión más pulida y definitiva. Cada versión se probaba por los miembros del equipo, y se sacaban diferentes conclusiones de mejora para futuras modificaciones.

Algunas de estas correcciones que se llevaron a cabo fueron mejoras en la interfaz de usuario, como por ejemplo, cambios en los márgenes para que se viera bien la aplicación en todos los dispositivos; incorporación de nuevas funcionalidades que necesitaba la aplicación para tener una mejor experiencia durante su uso, como el hecho de añadir diferentes botones para volver atrás; y además, mejorar diferentes problemas relacionados con rendimiento y con el manejo de errores.

Una vez que la APK estuvo en una etapa suficientemente avanzada, decidimos distribuirla a un grupo de usuarios. La idea principal de esto fue permitir a otros usuarios probar la aplicación en condiciones reales y obtener opiniones detalladas sobre su experiencia. Este proceso fue importante para recibir sugerencias sobre posibles mejoras y nuevas funcionalidades.

7.2. Funcionalidades evaluadas

El test se ha diseñado con el objetivo de obtener una visión general sobre la usabilidad y funcionalidad de la aplicación desde el punto de vista de los usuarios finales. De esta manera es posible identificar aspectos a mejorar para posteriormente implementar cambios que hagan la aplicación más deseable para el usuario objetivo.

Se ha buscado que las preguntas del test abarquen todos los aspectos de la aplicación, desde características de usabilidad hasta utilidad de las funcionalidades concretas de la aplicación. A continuación se enumeran más en detalle las características concretas evaluadas en el cuestionario:

- Diseño de la interfaz
- Usabilidad del menú
- Registro de un usuario
- Creación de dieta
- Visualización de las comidas diarias
- Lista de la compra
- Seguimiento de peso
- Modificación de un perfil

Para abarcar todas estas cuestiones se diseñó un test de 12 preguntas, en el que se le pide al usuario que valore del 1 al 5 en qué grado de acuerdo está con la afirmación, siendo 1 muy en desacuerdo y 5 muy de acuerdo. A continuación se detallan las preguntas del test:

1. La interfaz es atractiva
2. Los menús de la aplicación me han permitido acceder fácilmente a todas las características
3. La aplicación es fácil de usar
4. La aplicación me permite crear un usuario acorde a mis preferencias
5. La aplicación me ha permitido personalizar satisfactoriamente mi perfil
6. La aplicación me permite crear dietas adaptadas a mis necesidades
7. La calidad y viabilidad de las comidas sugeridas es suficiente
8. Las comidas sugeridas cumplen con las restricciones impuestas
9. Las instrucciones e ingredientes de cada comida son útiles
10. La lista de la compra me facilita la organización de las comidas
11. El seguimiento de peso me ayuda a mantener un control regular de mi peso
12. El seguimiento de peso me ayuda a visualizar mejor mi progreso de cara al objetivo de mi dieta

Esta selección de preguntas nos ha permitido abarcar todos los aspectos de la aplicación, obteniendo la retroalimentación que necesitábamos para sacar conclusiones acerca de la utilidad general de la aplicación.

7.3. Resultado de la evaluación

A continuación se extraen conclusiones sobre los resultados obtenidos.

1. La interfaz es atractiva

10 respuestas

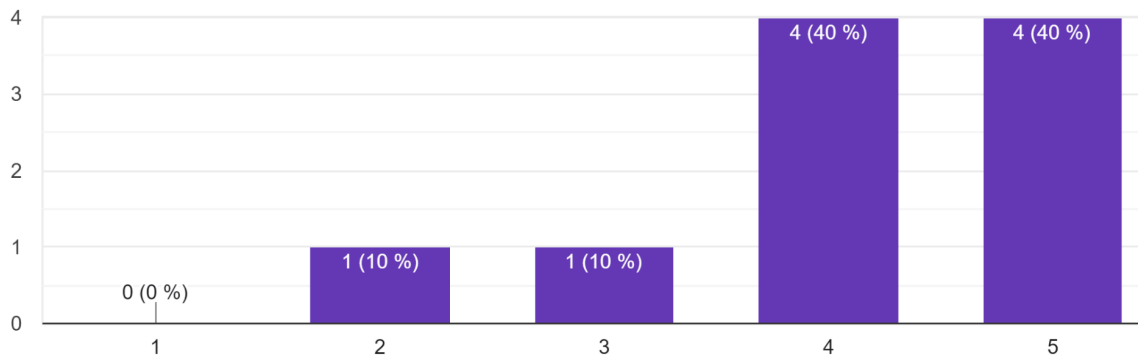


Figura 7.1: Resultados de la pregunta 1

1. Podemos observar en la figura 7.1 que la valoración de la interfaz de la aplicación es mayoritariamente buena, lo que nos hace ver que se ha cumplido el objetivo de desarrollar una aplicación llamativa desde el punto de vista del usuario.

2. La aplicación es fácil de usar

10 respuestas

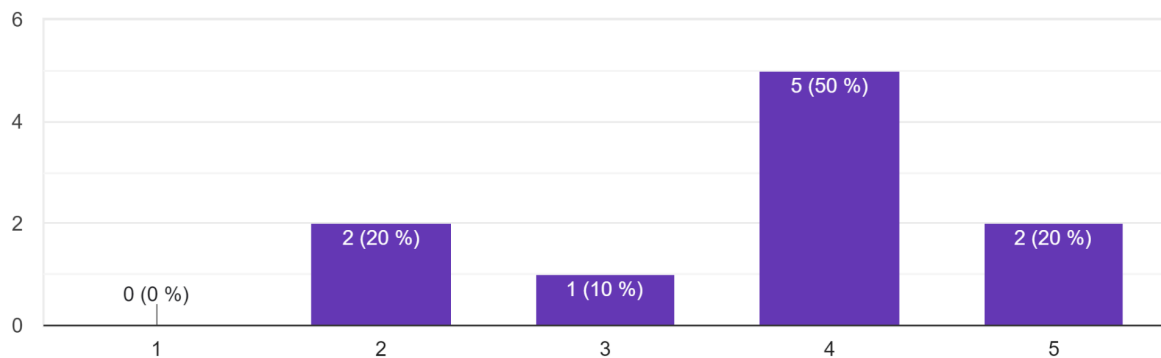


Figura 7.2: Resultados de la pregunta 2

2. Los resultados obtenidos (ver figura 7.2) en la pregunta sobre facilidad de uso de la aplicación son también satisfactorios, podemos observar que la mayoría de los usuarios encuentran la aplicación fácil de usar.

3. La aplicación me permite crear un usuario acorde a mis preferencias
10 respuestas

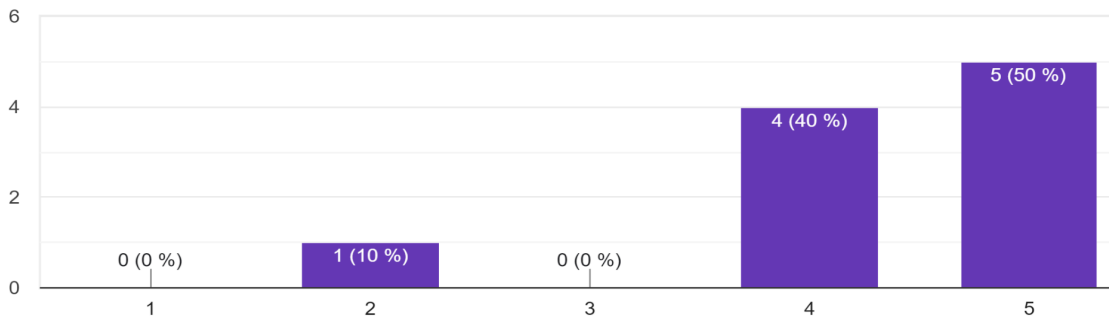


Figura 7.3: Resultados de la pregunta 3

3. Con esta pregunta se busca comprobar que los usuarios encuentran útiles las funcionalidades desarrolladas en cuanto a opciones de creación de un perfil (nombre de usuario, posibilidad de elegir avatar y añadir una contraseña...). Los resultados (ver figura 7.3) nos hacen interpretar que efectivamente estas opciones son útiles para la mayoría.

4. Los menús me han permitido acceder fácilmente a todas las características

10 respuestas

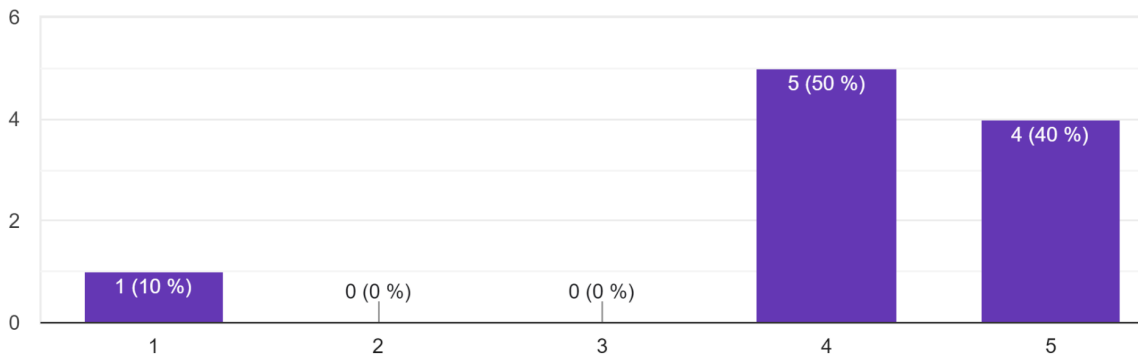


Figura 7.4: Resultados de la pregunta 4

4. En este apartado hemos obtenido unos resultados bastante satisfactorios (ver figura 7.4), esto se puede deber a que en todo el desarrollo se hizo especial hincapié en el desarrollo de menús con fácil usabilidad y accesibilidad a los recursos y funcionalidades de la aplicación.

5. La aplicación me ha permitido personalizar satisfactoriamente mi perfil

10 respuestas

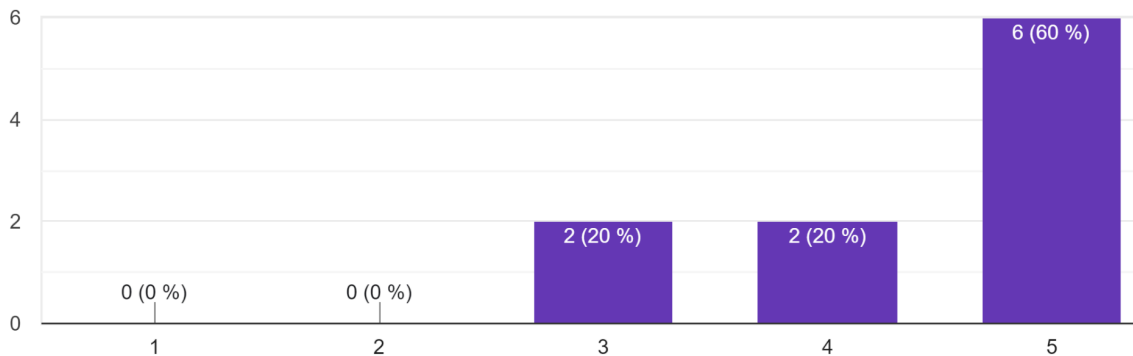


Figura 7.5: Resultados de la pregunta 5

5. Podemos observar que la mayoría de usuarios están satisfechos con la personalización de su perfil una vez creado (ver figura 7.5), sin embargo hay usuarios que han valorado menos positivamente este aspecto, lo que nos hace pensar que han echado en falta alguna característica adicional.

6. La aplicación me permite crear dietas adaptadas a mis necesidades

10 respuestas

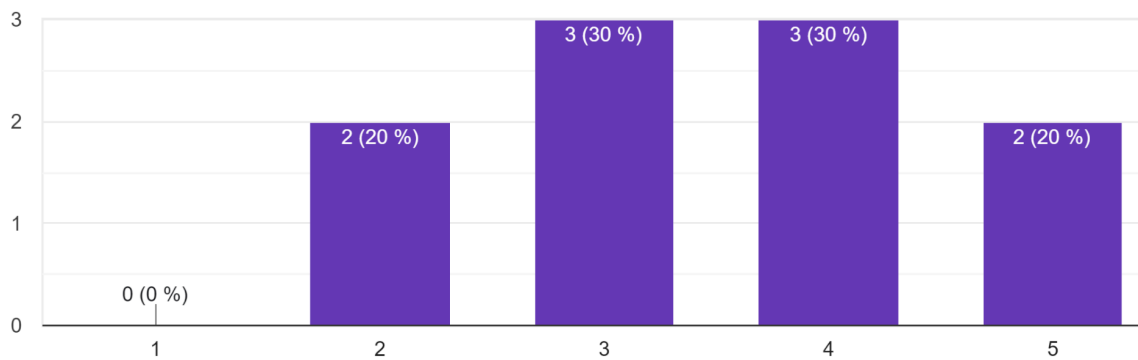


Figura 7.6: Resultados de la pregunta 6

6. En las respuestas a esta pregunta (ver figura 7.6) encontramos diversidad de opiniones, este hecho indica que el objetivo de crear una aplicación con facilidades para diferentes perfiles alimenticios no se ha terminado de completar.

7. La calidad y viabilidad de las comidas sugeridas es suficiente

10 respuestas

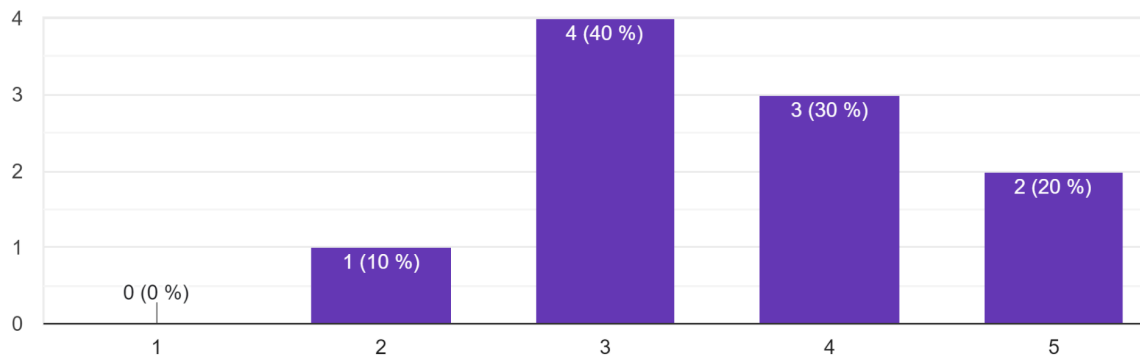


Figura 7.7: Resultados de la pregunta 7

7. Observamos que los usuarios en este caso han valorado diversamente la calidad de las comidas sugeridas para sus menús diarios (ver figura 7.7). Esto nos hace pensar que debemos dedicar recursos a mejorar las sugerencias de comida, tanto en calidad como en viabilidad.

8. Las comidas sugeridas cumplen con las restricciones impuestas

10 respuestas

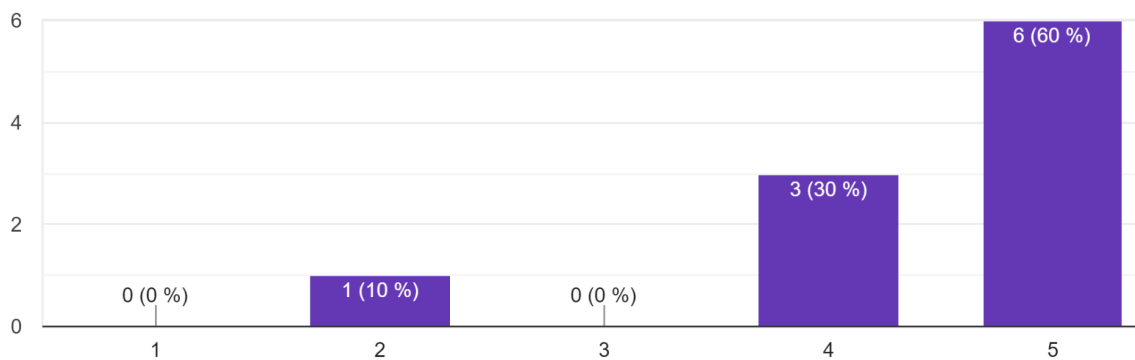


Figura 7.8: Resultados de la pregunta 8

8. Los resultados de esta pregunta (ver figura 7.8) nos confirman que los usuarios han podido excluir de su dieta ciertos alimentos, así como elegir un tipo concreto de dieta satisfactoriamente por lo que el objetivo de que la aplicación permita al usuario evitar ciertos alimentos en su dieta ha sido completado.

9. Las instrucciones e ingredientes de cada comida son útiles

10 respuestas

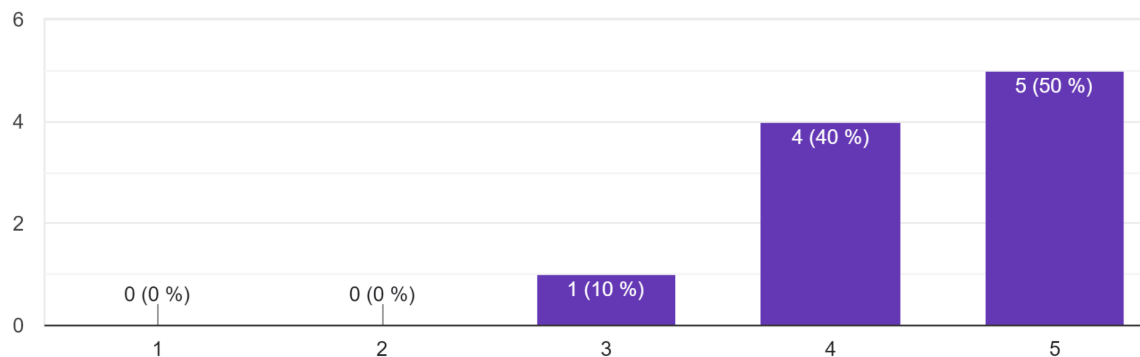


Figura 7.9: Resultados de la pregunta 9

9. Con esta pregunta pretendíamos evaluar la utilidad de los ingredientes e instrucciones de cada comida sugerida. Podemos observar (ver figura 7.9) que la mayoría de las respuestas son muy positivas lo que nos hace pensar que este aspecto de la aplicación cumple con sus expectativas y les resulta útil.

10. La lista de la compra me facilita la organización de las comidas

10 respuestas

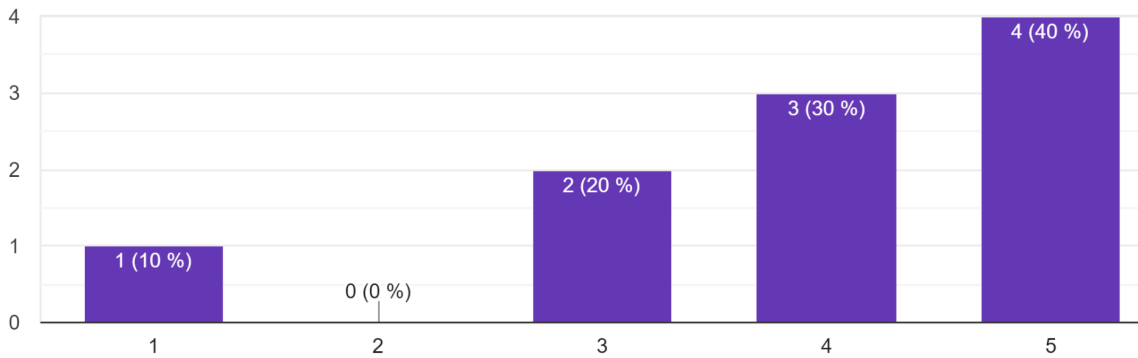


Figura 7.10: Resultados de la pregunta 10

10. A través del gráfico anterior (ver figura 7.10) podemos observar que los usuarios se sienten satisfechos con la funcionalidad de la lista de la compra. Entendemos que esta cumple con los requisitos impuestos y que facilita la administración de los ingredientes.

11. El seguimiento de peso me ayuda a mantener un control regular de mi peso

10 respuestas

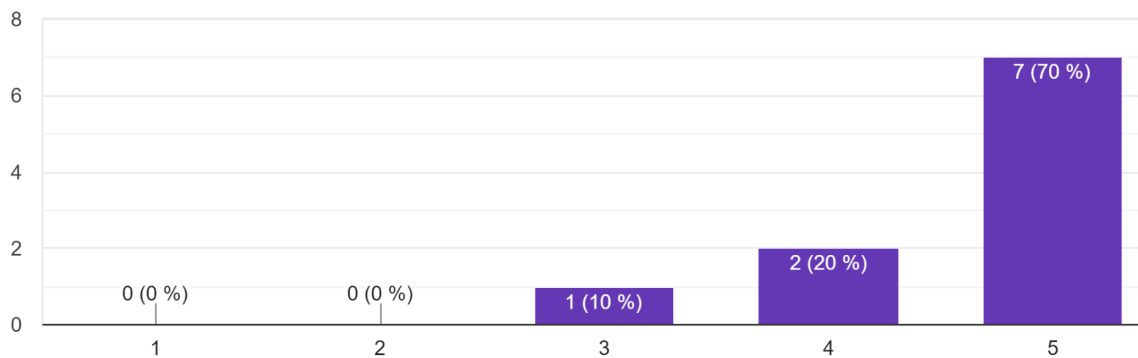


Figura 7.11: Resultados de la pregunta 11

11. Los resultados de esta pregunta han sido todo un éxito. El 70% de los usuarios (ver figura 7.11) encuentra útil la funcionalidad de seguimiento de peso, considerando que les ayuda a mantener un control de su peso.

12. El seguimiento de peso me ayuda a visualizar mejor mi progreso de cara al objetivo de mi dieta
10 respuestas

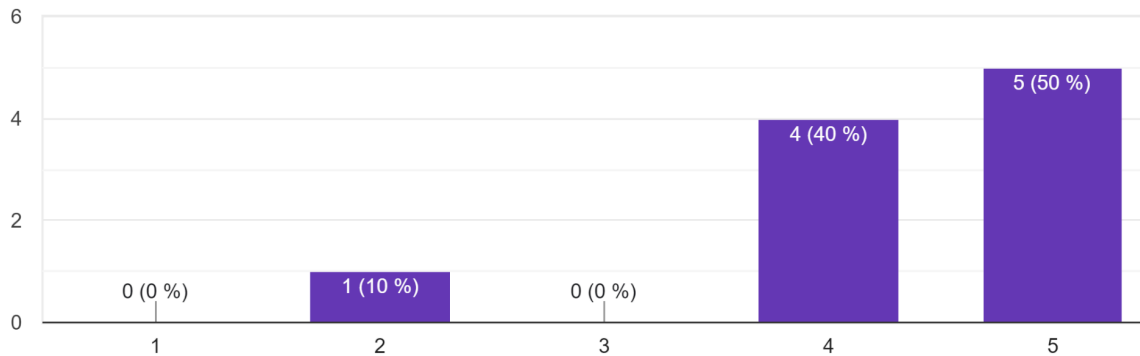


Figura 7.12: Resultados de la pregunta 12

12. Observamos que en general la respuesta de los usuarios es bastante buena (ver figura 7.12), estos han valorado positivamente la posibilidad de ver su progreso hacia su objetivo reflejado en las gráficas de peso.

Capítulo 8: Conclusiones y trabajo futuro

A continuación, se va a explicar las conclusiones que se han sacado en la realización de la aplicación y el trabajo de futuro que se tiene pensado realizar.

8.1. Conclusiones

El objetivo principal de desarrollar una aplicación Android para la creación de dietas y seguimiento de peso se ha cumplido con éxito. La aplicación proporciona una plataforma funcional para usuarios interesados en gestionar su dieta y peso de manera eficiente.

La aplicación desarrollada propone menús diarios de comida saludable adecuados para distintos tipos de dietas, de esta manera se ha conseguido una contribución significativa al campo de la salud y el bienestar, proporcionando a los usuarios una herramienta accesible y práctica para mejorar sus hábitos alimenticios y controlar su peso.

Durante las pruebas y evaluaciones realizadas, se observó que la aplicación es intuitiva y fácil de usar. Los usuarios pueden crear y seguir sus dietas con comodidad, lo que aumenta la probabilidad de adherencia a un plan alimenticio.

Los resultados preliminares de las pruebas de la aplicación muestran una respuesta positiva por parte de los usuarios beta, entre los que se incluyen familiares y amigos cercanos, quienes no solo han probado la aplicación, sino que también han aportado valiosas opiniones y sugerencias. Su participación activa ha permitido una mayor conciencia sobre las elecciones alimenticias y una mejor gestión del peso corporal.

A pesar del éxito inicial, se identificaron algunas limitaciones, como la falta de ciertas características avanzadas como por ejemplo una mayor personalización de la

aplicación o un posible foro en donde publicar los progresos o metas alcanzadas, y la necesidad de optimizar el rendimiento de la aplicación en dispositivos con recursos limitados.

8.2. Trabajo Futuro

Se debe trabajar en la mejora de las características existentes de la aplicación, como la inclusión de un mayor número de alimentos y recetas, extrayendo más datos de otras APIs, así como la incorporación de herramientas de análisis nutricional más detalladas.

Se podría implementar una función de personalización más avanzada que permita a los usuarios adaptar las dietas según sus preferencias alimenticias, como añadir dietas específicas en función de distintas patologías como una diabetes o hipertensión, filtrando en la API y añadiendo una consulta que saque productos adecuados a la selección marcada.

Se puede explorar la integración de tecnologías emergentes, como la inteligencia artificial o el aprendizaje automático, para ofrecer recomendaciones más precisas y personalizadas basadas en el progreso y las preferencias individuales de los usuarios. Por ejemplo usando una asistencia virtual y a través de un bot realizar distintas dudas o preguntas que este pueda ir respondiendo automáticamente.

Además del seguimiento del peso corporal, se podría considerar la incorporación de funciones para monitorizar otros aspectos de la salud, como la ingesta de agua, el nivel de actividad física y los patrones de sueño.

Es importante continuar realizando pruebas y recopilando información de los usuarios para identificar áreas de mejora y asegurar que la aplicación siga siendo relevante y útil para su audiencia objetivo.

Capítulo 9: Conclusions and future work

Next, the conclusions that have been drawn in the implementation of the application and the future work that is planned to be carried out will be explained.

9.1. Conclusions

The main objective of developing an Android application for diet creation and weight tracking was successfully achieved. The application provides a functional platform for users interested in effective diet and weight management.

The developed application proposes daily menus of healthy foods suitable for different types of diets, thus making a significant contribution to the field of health and wellness by providing users with an accessible and practical tool to improve their eating habits and control their weight.

During the tests and evaluations conducted, the application was found to be intuitive and easy to use. Users can conveniently create and follow their diets, increasing the likelihood of adherence to a diet plan.

Preliminary results from application testing show a positive response from beta users, including close family and friends, who not only tested the application but also provided valuable opinions and suggestions. Their active participation has led to greater awareness of dietary choices and better weight management.

Despite the initial success, some limitations were identified, such as the lack of certain advanced features, such as greater customization of the application or a possible forum to publish progress or goals achieved, and the need to optimize the performance of the application on devices with limited resources.

9.2. Future Work

Improve the existing functionality of the application, such as including a greater variety of foods and recipes, extracting more data from other APIs, and incorporating more detailed nutritional analysis tools.

Implement a more advanced customization feature that allows users to tailor diets according to their dietary preferences, such as adding specific diets based on different pathologies like diabetes or hypertension, by filtering in the API and adding a query to extract suitable products based on the selection made.

Explore the integration of emerging technologies, such as artificial intelligence or machine learning, to provide more precise and personalized recommendations based on the user's progress and individual preferences. For example, using a virtual assistant and bot to automatically answer various doubts or questions.

In addition to tracking body weight, consideration could be given to incorporating features to monitor other aspects of health, such as water intake, physical activity levels, and sleep patterns.

Continuing to test and gather user feedback is important to identify areas for improvement and ensure that the application remains relevant and useful to its target audience.

Contribuciones personales

Contribuciones de Antonio Bonillo Saiz

Lo primero que hice, y esto es algo que realizamos los 4 compañeros, fue recopilar información sobre aplicaciones similares a la que íbamos a desarrollar. Esto lo hicimos con el fin de encontrar ideas y funcionalidades útiles para implementar en nuestra aplicación, pero también con el fin de encontrar algo nuevo que aportar a los usuarios de aplicaciones de este tipo.

Lo siguiente fue familiarizarse con las herramientas de trabajo, Android Studio y desarrollo de aplicaciones Android. Ya que utilizamos Java, para mí no fue demasiado complicado acostumbrarme, sin embargo en el desarrollo de aplicaciones existen ciertas peculiaridades que sí me supusieron inicialmente un reto mayor.

Una vez definidas las funcionalidades que queríamos implementar en nuestra aplicación, decidimos que lo primordial era tener los datos fisiológicos del usuario y generar una dieta. Para ello lo primero que hice fue crear las pantallas de recopilación de datos, así como el almacenamiento interno de estos datos. En estas pantallas el usuario puede introducir su edad, peso, altura y ejercicio físico semanal, estos datos se mostrarán en los campos para rellenar si ya los ha introducido previamente. Estos datos se utilizan para calcular las necesidades calóricas personalizadas del usuario y luego ajustar la ingesta diaria de kilocalorías según el objetivo de dieta que eligiese el usuario (pérdida de peso, mantenimiento o ganancia de masa muscular).

Lo siguiente a realizar era crear una estructura de base de datos capaz de relacionar usuarios y dietas. Para ello añadí inicialmente varias tablas como la de usuario, dieta, comida y una tabla auxiliar que relaciona para cada día de la dieta del usuario, los ids de las comidas asignadas así como los macronutrientes.

Una vez teníamos una estructura de base de datos definida y funcional, programé el sistema de login y logout implementado internamente como una cookie de sesión y presentado al usuario como la posibilidad de tener hasta 6 usuarios diferentes con sus respectivas dietas. También desarrollé la pantalla de creación de usuarios, que permite crear un usuario con nombre, contraseña (opcional) y avatar. Tras esto, implementé también el sistema de borrado de usuarios.

En el momento en el que ya existe la posibilidad de crear una dieta y un usuario pero no visualizar la dieta en sí, lo conveniente es diseñar una manera amigable para el usuario que le permita ver las comidas de cada día. Para ello añadí una pantalla de visualización de la dieta, en la que se muestra un calendario en el que al pulsar en un día, te muestra las comidas diarias. Para ello utilicé la tabla de base de datos que relacionaba las comidas del día para cada día de la dieta, y que he mencionado anteriormente.

La funcionalidad del seguimiento de peso ha sido sin duda la más desafiante de mis aportaciones al proyecto, implementar un apartado que permitiese al usuario visualizar su peso gráficamente y por fechas fue desde el principio una funcionalidad muy complicada. El trayecto comenzó con labores de investigación junto a mis compañeros en busca de una biblioteca ligera y sencilla de usar que permitiese generar gráficas. Encontramos la biblioteca jplot [\[30\]](#) y MPAndroidChart [\[31\]](#), ambas son bibliotecas de creación de gráficos muy completas, sin embargo, tenían demasiadas funcionalidades y no cumplían con nuestro requisito de biblioteca ligera. Tras haber investigado un poco más, encontré la biblioteca GraphView [\[32\]](#). Esta librería es perfecta para lo que buscamos y relativamente ligera y sencilla de usar.

Tras haber probado algún gráfico, y una vez entendido el funcionamiento de la librería, comencé a desarrollar la parte de seguimiento de peso de la aplicación. Lo primero que necesitábamos era un apartado donde introducir el peso diariamente, decidí que ya que tenemos un calendario, lo mejor sería permitir que el usuario elija

entre ver/editar el peso del día y mostrar sus comidas para ese mismo día. Por lo tanto, fue lo primero que implementé, añadiendo a la base de datos la tabla de seguimiento de peso, que relaciona peso, usuario y día. De esta manera tenía la información del peso del usuario organizada para poder manipularla fácilmente y generar una gráfica de puntos.

Cuando ya implementé la generación de una línea de puntos con los pesos diarios registrados, lo siguiente fue darle la posibilidad al usuario de visualizar en la gráfica solamente los pesos para los días pertenecientes al periodo de los últimos 7 días, al último mes o mostrar todos los datos de peso que haya registrado. Esto lo implementé a través de un spinner con las opciones mencionadas.

El siguiente paso fue añadir una serie de puntos diferenciada de la del peso del usuario, que representase una estimación del peso óptimo que debía conseguir el usuario variando para cada día, según su objetivo de peso de la dieta. Por último, añadí una pequeña leyenda para dar contexto a ambas series de datos y un botón que permite al usuario ocultar/mostrar la serie de datos de peso óptimo.

Fui el encargado de confeccionar el test de evaluación en usuarios, elaborando las preguntas de este así como decidiendo qué aspectos de la aplicación se iba a evaluar. También me he encargado de recopilar y analizar los resultados obtenidos, redactando conclusiones sobre cada una de las preguntas que se le han formulado a los usuarios, extrayendo aspectos a mejorar o características que se han implementado con éxito.

Como última contribución, he redactado una serie de apartados y capítulos de esta memoria que a continuación se enumeran: los capítulos 1 y 3 en su totalidad, el apartado 2.3, 5.1, 5.2, 5.3, 5.6, 5.7, 6.1, 7.2 y 7.3.

Contribuciones de Blanca García Álvarez

En mi contribución personal para la memoria de mi Trabajo de Fin de Grado (TFG), mi aporte se centró en la implementación de diversas funcionalidades y en la mejora general de la experiencia del usuario. Este proceso involucró un análisis completo de aplicaciones similares, así como la descarga y estudio detallado de una de ellas para comprender su funcionamiento y adquirir ideas para nuestra propia aplicación.

Una de las características principales que desarrollé fue un sistema de consejos dinámicos basados en la base de datos. Este sistema permite que los usuarios reciban recomendaciones personalizadas según el tipo de dieta que están siguiendo. Por ejemplo, se generan consejos específicos para objetivos como ganar masa muscular, perder peso o mantenerse saludable, además de consejos generales aplicables a cualquier tipo de dieta. Esta funcionalidad no solo enriquece la experiencia del usuario al proporcionar información relevante y útil, sino que también anima y fomenta hábitos alimenticios más saludables y adaptados a las necesidades individuales de cada usuario.

Luego me dediqué al desarrollo de la vista "Summary" de la aplicación. Esta vista proporciona una experiencia centralizada y eficiente para los usuarios. Actúa como un punto de acceso rápido a algunas de las funcionalidades clave de la aplicación, ofreciendo una visión general de la dieta y brindando consejos personalizados para mejorar los hábitos alimenticios.

Principalmente, presenta una sección dedicada a los consejos, los cuales se adaptan según el tipo de dieta del usuario. Estos consejos proporcionan orientación y recomendaciones específicas para ayudar al usuario a alcanzar sus objetivos de salud.

En esta vista se incluyen varias frases a modo de enlace, lo que significa que al pinchar en las distintas frases, se le redirige a los distintos fragmentos.

Una de ellas, permite al usuario acceder directamente al menú correspondiente al día actual, por lo que se le redirige al fragmento del menú de hoy diseñado específicamente para su perfil de dieta. Y la otra, permite al usuario cambiar de dieta fácilmente, simplemente pinchando sobre la frase, le lleva al fragmento de configuración (settings) donde puede realizar dicho cambio de manera rápida y sencilla.

También está formada por una frase que especifica al usuario, el tipo de dieta que ha elegido, lo que le ayuda a recordar y mantenerse enfocado en sus objetivos de salud.

Para mejorar la presentación visual y la interactividad de la aplicación, utilicé bibliotecas como Spannable y StyleSpan, que permiten resaltar ciertas partes del texto y crear una experiencia de usuario más atractiva y agradable. Además, también cuenta con una imagen generada por inteligencia artificial para mejorar la estética y la experiencia visual de la vista summary. [\[33\]](#)

Cabe destacar que cuando un usuario no tiene una dieta creada, esta vista cuenta con una frase motivacional que lo alienta a crear una dieta para aprovechar al máximo la experiencia de la aplicación. Esto ayuda a fomentar la participación del usuario y a garantizar que aproveche todas las funcionalidades disponibles para mejorar su salud y bienestar.

En cuanto al diseño de la aplicación, dediqué tiempo a experimentar con varias combinaciones de colores y estilos para crear una interfaz estéticamente atractiva y coherente. Además, optimicé la navegación y la disposición de los elementos de la interfaz de usuario para garantizar una experiencia fluida y fácil de usar para los usuarios.

Dentro del marco estructural de la memoria, realicé la redacción de los capítulos 8 y 9 de la memoria del TFG.

Me enfoqué en las conclusiones del proyecto y en el trabajo futuro. Aquí, analicé los logros alcanzados durante el desarrollo de la aplicación, así como las áreas que podrían mejorarse en futuras iteraciones. Las posibles propuestas para la expansión de funcionalidades, la integración de tecnologías emergentes y la mejora continua de la experiencia del usuario.

Contribuciones de Víctor Ortego Martín

Una de las primeras tareas que realicé fue la búsqueda de diferentes aplicaciones similares al tema que íbamos a tratar. Cada miembro se encargó de buscar aplicaciones para después ponerlas en común y determinar qué aspectos positivos podíamos coger de cada una de ellas para realizar la mejor aplicación posible.

Después de familiarizarme con el entorno y con las tecnologías que íbamos a utilizar, me dediqué a investigar diferentes API sobre recetas y comidas, para posteriormente utilizarla en el proyecto como base de datos. Tras una búsqueda exhaustiva, me encargué de realizar comparaciones entre las posibles candidatas para determinar cuál era la mejor opción y cuál nos podía aportar más al proyecto. Fue satisfactorio el hecho de investigar este tipo de API, ya que desconocía su existencia, y me pareció muy interesante entender cómo funcionan y qué aportaba cada una de ellas.

Una vez elegida la API que íbamos a utilizar, al ser la persona que se había encargado de investigar acerca de las API, también me encargué de integrar la API en el proyecto, e implementar varios métodos que íbamos a necesitar en un futuro con diferentes peticiones a la API, como por ejemplo, la obtención de la información de una receta, o la petición de realizar un menú diario.

El hecho de que nunca hubiera trabajado con una API en un proyecto, hizo que me llevara mucho trabajo entender cómo integrar la API, y realizar los diferentes métodos y peticiones a la misma.

Después de realizar la integración de la API, me dediqué a realizar diferentes pruebas para ver su correcto funcionamiento. Una de esas pruebas consistió en realizar diferentes peticiones y mostrar por la consola los resultados adquiridos. Tras ver que esto funcionaba correctamente, el siguiente paso fue investigar y determinar cómo iba

a mostrar esa información en la aplicación. Para ello, investigué acerca de las interfaces que se podían realizar en Android, y tras varios prueba y error, encontré la manera de mostrar en una interfaz de usuario la información, en este caso, de una receta. Dicha interfaz se quedó en el proyecto como ejemplo para futuras interfaces a desarrollar.

Tras ver el correcto funcionamiento de la API, me dediqué a realizar una interfaz donde se iba a mostrar las tres recetas que tenía el usuario en un día, es decir, desayuno, comida y cena. Y posteriormente, la interfaz encargada de mostrar cada una de las recetas de forma individual, con su foto, su descripción, sus ingredientes y sus instrucciones. Realizar estas interfaces me resultó un poco complicado al principio, ya que las interfaces de Android no son tan sencillas de realizar, y tuve que investigar mucho para realizar unas interfaces competentes.

Posteriormente, dediqué mi tiempo a realizar un modelo de datos para poder almacenar la información de los ingredientes, y relacionar esos ingredientes con sus recetas. Esto lo hice para poder implementar después una interfaz donde se podría visualizar una lista de la compra con los ingredientes que necesita el usuario para realizar las recetas asignadas en su dieta.

Esta interfaz diferenciaba los ingredientes según si eran necesarios para el desayuno, la comida o la cena, y además, el usuario puede marcar o desmarcar cada ingrediente para confirmar si posee dicho ingrediente o no. Y para mejorar el filtrado de los ingredientes, también implementé dos filtros, uno donde diferencia entre ingredientes marcados o no, y otro donde muestra solamente los ingredientes hasta un rango de días determinado, por ejemplo, los ingredientes de las recetas de los próximos 5 días consecutivos.

Tras realizar esta implementación, la cual, me llevó bastante tiempo, ayudé a Jose a realizar la implementación de poder modificar los datos de un usuario. Entre los

dos, terminamos el desarrollo de la funcionalidad, realizando diferentes pruebas para comprobar su correcto funcionamiento.

Después, me ofrecí a implementar la funcionalidad de generar un nuevo menú del día, en caso de que el usuario no estuviera satisfecho con su menú actual. Este desarrollo no fue muy complicado, gracias a haber realizado funcionalidades similares en el proyecto, pero sí que llevó bastante tiempo realizarla por la lógica que conllevaba.

Casi al finalizar el proyecto, nos dimos cuenta de que la función de introducir comidas que no se quieren en la dieta, era difícil de interpretar, por lo que decidí implementar un autocompletar, en el que introduciendo un ingrediente, te mostraba varios ingredientes similares al introducido, para seleccionarlo de una manera más sencilla.

Ya por último, me dediqué a realizar pequeñas correcciones en el proyecto, y además, desarrollé algunos puntos de la memoria, como los apartados 1 y 2 del capítulo 2, el apartado 3 del capítulo 3, el apartado 1 del capítulo 7, parte del capítulo 5 y otros aspectos como el resumen.

Contribuciones de Jose Ángel Suárez Pesquera

En mi contribución personal para la memoria de mi Trabajo de Fin de Grado (TFG), llevé a cabo una comparación exhaustiva con otras aplicaciones de dietas con el objetivo de obtener inspiración y perspectiva. Durante la fase inicial del proyecto, me dediqué a aprender el manejo de Android Studio a través de tutoriales especializados. Una vez familiarizado con la interfaz, procedí a implementar diversas funcionalidades clave.

Durante la fase de desarrollo, realicé la creación del menú principal, llamada en la aplicación, HomeActivity, con un enfoque especial en la identidad visual de la aplicación. Este menú, cuyo diseño pensé hacerlo de manera distintiva, incorporando el logo característico y organizándose de forma eficiente mediante una barra de navegación que facilita el acceso a sus diversos fragmentos. Me enfoqué en dar especial atención a la experiencia del usuario, asegurándome de que la navegación resultara intuitiva y atractiva.

En paralelo, trabajé en el diseño y la implementación de la página de ajustes de la aplicación. Este componente no solo ofrece funcionalidades clave, como la eliminación de una dieta, que fue bastante fácil de implementar pero a su vez es primordial, sino que también se destacó por integrar la política de privacidad correspondiente. La transparencia y el cuidado de la privacidad del usuario fueron consideraciones fundamentales en este proceso.

Dentro de la página de los ajustes, está la implementación de la función de modificación de perfil. Este componente es básico para personalizar la experiencia del usuario, y se logró con la colaboración de Victor, quien aportó contribuciones para perfeccionar los detalles finales.

La colaboración en este aspecto garantizó que la función cumpliera con los estándares de calidad y usabilidad establecidos para el proyecto.

Por último, destaco el desarrollo del sistema de notificaciones de la aplicación como una tarea significativa y compleja en el proceso. La implementación de este componente fue particularmente desafiante, ya que, tras una exhaustiva investigación, se encontró que la integración con Firebase de Google no resultaba viable para nuestro proyecto. Superando este obstáculo, se exploraron soluciones alternativas que se ajustaran a nuestras necesidades, consiguiendo así lo que queríamos con notificaciones personalizadas para nuestros usuarios.

Dentro del marco estructural de la memoria, realicé la redacción de varias introducciones, abarcando diversos aspectos del proyecto.

El Capítulo 4, focalizado en los casos de uso de la aplicación, se convirtió en un componente esencial de la memoria. En este capítulo, analicé exhaustivamente todos los escenarios y situaciones en los que los usuarios interactúan con la aplicación asegurándome de que todos los requisitos planteados inicialmente fueran plenamente contemplados.

Para proporcionar una perspectiva más clara y visual de estas funcionalidades, incorporé un pequeño diagrama. Este elemento gráfico no solo sirve como un recurso visual atractivo, sino que también actúa como una herramienta efectiva para comprender la interrelación y la cobertura de los casos de uso dentro de la aplicación.

La inclusión de este diagrama busca mejorar la comprensión del lector y facilitar la asimilación de la complejidad inherente a la aplicación.

Finalmente, una vez que terminamos todas nuestras tareas, me ofrecí a ser el encargado de lanzar la APK (Android Application Package) para poder probar la aplicación en nuestros respectivos móviles. Además, en este proceso añadí nuestro logo de la aplicación para que fuera el icono que apareciera en el móvil cuando te la descargas.

Por otro lado, nuestro profesor nos comentó que sería bueno añadir a la memoria diferentes capturas de pantalla de las actividades de la aplicación más destacables. Todo ello, se encuentra en el punto 5 en el cual hago todas las capturas de pantalla y la redacción de los subapartados 5.3, 5.4, 5.5 y 5.6. Además, he realizado el diagrama de entidad-relación de nuestra aplicación usando virtualchard, una gran aplicación para la realización de diagramas de todo tipo al igual que los diagramas de casos de uso.

Bibliografía

- [1] Marqueta de Salas, M., Martín-Ramiro, J. J., Rodríguez Gómez, L., Enjuto Martínez, D., & Juárez Soto, J. J. (2016). Hábitos alimentarios y actividad física en relación con el sobrepeso y la obesidad en España. *Revista española de nutrición humana y dietética*, 20(3), 224-235. Disponible en:
https://scielo.isciii.es/scielo.php?pid=S2174-51452016000300009&script=sci_arttext
- [2] FoodData Central. Disponible en:
<https://fdc.nal.usda.gov/>
- [3] PlanEat. Disponible en:
<https://planeat.app/>
- [4] 8Fit. Disponible en:
<https://8fit.com/es/>
- [5] MyFitnessPal. Disponible en:
<https://www.myfitnesspal.com/es>
- [6] Halen. Disponible en:
<https://www.halenapp.com/>
- [7] MyFoodPlan. Disponible en:
<https://lanzadera.es/proyecto/my-food-plan/>

[8] Apilayer. Disponible en:

<https://apilayer.com/>

[9] Spoonacular Application Programming Interface. Disponible en:

<https://spoonacular.com/food-api>

[10] Nutritionix. Disponible en:

<https://www.nutritionix.com/>

[11] Nutritics Application Programming Interface. Disponible en:

<https://www.nutritics.com/en/services/api>

[12] ReciPal. Disponible en:

<https://www.recipal.com/>

[13] FatSecret Platform Application Programming Interface. Disponible en:

<https://platform.fatsecret.com/platform-api>

[14] LogMeal Application Programming Interface. Disponible en:

<https://www.logmeal.es/>

[15] Chompthis. Disponible en:

<https://chompthis.com/>

[16] BigOven Application Programming Interface. Disponible en:

<https://api2.bigoven.com/>

[17] TheMealDB. Disponible en:

<https://www.themealdb.com/>

[18] Zestful. Disponible en:

<https://zestfuldata.com/>

[19] Peralta, C. A. (2016). Elementos fundamentales en el cálculo de dietas. Editorial El Manual Moderno. Disponible en:

<https://books.google.es/books?hl=es&lr=&id=DQB8DQAAQBAJ&oi=fnd&pg=PP5&dq=c%C3%A1lculo+de+tasa+metab%C3%B3lica+basal&ots=HU0mBrHcZe&sig=brloNYyq-D S25AeNrArY-m1d2Y8#v=onepage&q=c%C3%A1lculo%20de%20tasa%20metab%C3%B3lica%20basal&f=false>

[20] Brad Schoenfeld. (2016). Ciencia y desarrollo de la hipertrofia muscular. Ediciones Tutor.

[21] Sanhueza Catalán, J., Durán Agüero, S., & Torres García, J. (2015). Los ácidos grasos dietarios y su relación con la salud. *Nutrición Hospitalaria*, 32(3), 1362-1375. Disponible en:

https://scielo.isciii.es/scielo.php?pid=S0212-16112015000900054&script=sci_arttext

[22] OH, H. CLASIFICACIÓN DE LOS GLÚCIDOS. Disponible en:

<https://uruguayeduca.anep.edu.uy/sites/default/files/inline-files/Gl%C3%BAcidos.pdf>

[23] Android Studio. Disponible en:

<https://developer.android.com/studio>

[24] Java (lenguaje de programación). Disponible en:

<https://www.java.com/es/>

[25] Spoonacular Documentation. Disponible en:

<https://spoonacular.com/food-api/docs>

[26] Dall-E. Disponible en:

<https://openai.com/index/dall-e/>

[27] Github. Disponible en:

<https://github.com/>

[28] DB Browser for SQLite. Disponible en:

<https://sqlitebrowser.org/>

[29] Google Fonts. Disponible en:

<https://fonts.google.com/>

[30] Johnspice. Disponible en:

<https://github.com/johnspice/jplot-android>

[31] PhilJay. Disponible en:

<https://github.com/PhilJay/MPAndroidChart>

[32] jjoe64. Disponible en:

<https://github.com/jjoe64/GraphView>

[33] freepik, Disponible en:

<https://www.freepik.es/ia/generador-de-imagenes>