

Memoria

FitApp: Aplicación para controlar el plan de nutrición y entrenamiento

Trabajo de Fin de Grado

Ingeniería Informática



**VNiVERSiDAD
D SALAMANCA**

Septiembre de 2023

Autor

María Pérez Morales

Tutor/a

André Filipe Sales Mendes

CERCIFICADO DEL TUTOR

D. André Filipe Sales Mendes, profesor/a del Departamento de Informática y Automática de la Universidad de Salamanca,

HACE CONSTAR:

Que el trabajo titulado “FitApp: Aplicación para controlar el plan de nutrición y entrenamiento”, que se presenta, ha sido realizado por María Pérez Morales, con DNI 03941068H y constituye la memoria del trabajo realizado para la superación de la asignatura Trabajo de Fin de Grado en Ingeniería Informática en esta Universidad.

Salamanca, 07 de septiembre de 2023 Fdo.:

RESUMEN

La memoria presenta la documentación completa de mi proyecto de fin de grado titulado "FitApp: Aplicación para controlar el plan de nutrición y entrenamiento". El propósito principal de este proyecto es desarrollar una aplicación móvil para la administración efectiva de planes de nutrición y entrenamiento, orientada para asistir al usuario en la incorporación de un estilo de vida saludable y activo. Esta aplicación, denominada FitApp, ha sido minuciosamente diseñada para garantizar su accesibilidad tanto en dispositivos móviles como en tabletas con sistema operativo Android.

FitApp ha sido concebida con el objetivo de proporcionar a los usuarios una herramienta integral que les permita supervisar y gestionar su vida diaria, abarcando tanto su consumo alimenticio como su actividad física. El enfoque es ayudar a los usuarios a alcanzar sus objetivos de salud al proporcionarles una serie de funciones clave.

En el ámbito de la nutrición, FitApp brinda a los usuarios la capacidad de registrar sus hábitos alimenticios y, a partir de esta información, ofrece recomendaciones dietéticas personalizadas basadas en sus preferencias. Los usuarios pueden optar por seguir una dieta sugerida por la aplicación o crear una propia. Además, la aplicación permite la gestión de la ingesta calórica diaria, lo que proporciona un mayor control sobre la alimentación.

En lo que respecta al ámbito deportivo, la aplicación ofrece a los usuarios la posibilidad de registrar sus sesiones de ejercicio de manera instantánea o manual, lo que les otorga la flexibilidad de ajustar los detalles de acuerdo a sus preferencias personales. Además, la aplicación calcula con precisión las calorías consumidas durante las actividades físicas realizadas a lo largo del día, proporcionando una visión completa del gasto calórico asociado a cada sesión deportiva.

Además, los usuarios pueden crear listas personalizadas de ejercicios y alimentos favoritos, agilizando la elección diaria. Esto ahorra tiempo y adapta la aplicación a las preferencias individuales, optimizando la experiencia del usuario.

FitApp se distingue de otras aplicaciones en el mercado, ya que une la gestión de la nutrición y el entrenamiento en una sola plataforma, permitiendo a los usuarios comprender y aprovechar la interacción entre estos dos aspectos esenciales para la salud.

La creación y desarrollo de FitApp ha sido fundamentada en tecnologías vanguardistas de la industria. Para dar vida a la interfaz y funcionalidades de la aplicación, se ha optado por la combinación de lenguaje de programación JavaScript y el framework Vue.js, cuya eficacia en la creación de interfaces interactivas es ampliamente reconocida. Además, se ha integrado Ionic.io, una plataforma de renombre destinada a la edificación y despliegue de aplicaciones móviles de última generación.

Para agilizar el proceso de desarrollo y optimizar la experiencia del usuario, se ha implementado el framework Vuetify. Este framework, construido sobre Vue.js, ha posibilitado la creación ágil de componentes altamente personalizables y la materialización de una estética atractiva y cohesiva en la interfaz de la aplicación. En consecuencia, FitApp no solo se beneficia de un enfoque tecnológico puntero, sino que también se enriquece con una experiencia de usuario fluida y visualmente agradable.

Tanto el almacenamiento y la gestión de los datos, como la autenticación de usuarios han sido gestionados mediante Firebase, una completa y avanzada plataforma de desarrollo ofrecida por Google. Firebase ha permitido garantizar la confidencialidad y la integridad de los datos de los

usuarios, mediante la implementación medidas de protección ofrecidas por la propia herramienta. Además, ha simplificado significativamente la administración de la aplicación desde una perspectiva técnica, al proporcionar servicios que abarcan desde el almacenamiento en la nube hasta la autenticación segura. En este contexto, el almacenamiento de datos hace referencia al proceso de guardar y recuperar información, mientras que la autenticación de usuarios se refiere a la validación de las identidades de los usuarios que acceden a la plataforma, garantizando su autorización y protegiendo la integridad de los datos sensibles.

La documentación del proyecto se estructura en tres secciones fundamentales: el presente documento, denominado "memoria", y seis anexos complementarios. Estos anexos proveen una amplia exposición de las estrategias y técnicas empleadas con el fin de alcanzar los objetivos trazados en el proyecto, incorporando enfoques característicos del Proceso Unificado, un marco de trabajo ampliamente reconocido en ingeniería de software. Además, se suministra documentación técnica que brinda una comprensión exhaustiva del código desarrollado, y un manual de usuario diseñado para dirigir de manera efectiva a los usuarios en la utilización de la aplicación resultante. Por otra parte, en la memoria se efectuará una exposición detallada de los conceptos teóricos fundamentales que han sentado las bases para la ejecución de este proyecto, junto con aspectos cruciales provenientes de los anexos, entre otros puntos relevantes que irán siendo mencionados a lo largo del documento.

Palabras clave: gestión de un plan de nutrición y entrenamiento, aplicación móvil, JavaScript, Vue.js, Ionic.io, Vuetify, Firebase, Proceso Unificado, manual de usuario.

SUMMARY

The report presents the complete documentation of the undergraduate thesis project titled 'FitApp: Application for Managing Nutrition and Training Plans.' The main purpose of this project is to develop a multifaceted platform for the effective management of nutrition and training plans, aimed at assisting the user in adopting a healthy and active lifestyle. This application, named FitApp, has been meticulously designed to ensure its accessibility on both mobile devices and tablets running the Android operating system.

FitApp has been conceived with the objective of providing users with a comprehensive tool that allows them to monitor and manage their daily lives, covering both their dietary intake and physical activity. The focus is to help users achieve their health goals by providing them with a set of key features.

In the realm of nutrition, FitApp provides users with the ability to record their eating habits and, based on this information, offers personalized dietary recommendations based on their preferences. Users can choose to follow an app-suggested diet or create their own. Additionally, the app allows for the management of daily calorie intake, providing greater control over eating habits.

Regarding the sports domain, the application offers users the option to instantly or manually record their exercise sessions, giving them the flexibility to adjust details according to their personal preferences. Furthermore, the app accurately calculates the calories burned during physical activities throughout the day, offering a complete view of the caloric expenditure associated with each workout session.

Moreover, users can create personalized lists of favorite exercises and foods, streamlining daily choices. This saves time and tailors the app to individual preferences, optimizing the user experience.

FitApp stands out from other apps in the market by combining nutrition and training management into a single platform, allowing users to understand and leverage the interaction between these two essential health aspects.

The creation and development of FitApp have been based on cutting-edge industry technologies. To bring the app's interface and functionalities to life, a combination of JavaScript programming language and the Vue.js framework has been chosen, widely recognized for its effectiveness in creating interactive interfaces. Additionally, Ionic.io, a renowned platform for building and deploying state-of-the-art mobile applications, has been integrated.

To streamline the development process and enhance the user experience, the Vuetify framework has been implemented. Built on Vue.js, this framework has enabled agile creation of highly customizable components and the realization of an attractive and cohesive aesthetic in the app interface. Consequently, FitApp not only benefits from a leading technological approach but also enriches the user experience with smooth and visually pleasing interactions.

Data storage and user authentication have been managed through Firebase, a comprehensive and advanced development platform offered by Google. Firebase has ensured the confidentiality and security of user data by implementing protective measures offered by the tool itself. Additionally, it has significantly simplified application management from a technical perspective by providing services ranging from cloud storage to secure authentication. In this context, data storage refers to the process of saving and retrieving information, while user

authentication pertains to the validation of identities of users accessing the platform, ensuring their authorization and safeguarding the integrity of sensitive data.

The project documentation is structured into three fundamental sections: this document, named 'report,' and six complementary annexes. These annexes provide an extensive exposition of the strategies and techniques employed to achieve the project's goals, incorporating characteristic approaches of the Unified Process, a widely recognized framework in software engineering. Furthermore, technical documentation is provided that offers a comprehensive understanding of the developed code, along with a user manual designed to effectively guide users in utilizing the resulting application. On the other hand, the report will provide a detailed exposition of the fundamental theoretical concepts that have laid the groundwork for the execution of this project, along with crucial aspects from the annexes, among other relevant points that will be mentioned throughout the document.

Keywords: nutrition and training plan management, mobile application, JavaScript, Vue.js, Ionic.io, Vuetify, Firebase, Unified Process, user manual.

ÍNDICE DE CONTENIDO

CERCIFICADO DEL TUTOR	2
RESUMEN.....	3
SUMMARY	5
ÍNDICE DE CONTENIDO.....	7
INDICE DE TABLAS	10
ÍNDICE DE ILUSTRACIONES.....	11
M.1 INTRODUCCIÓN.....	13
M.2	15
M.2.1 8fit	15
M.2.2 MyFitnessPal.....	16
M.3 OBJETIVOS	17
M.3.1 Objetivos del sistema	17
M.3.2 Objetivos personales	18
M.4 CONCEPTOS TEÓRICOS	19
M.4.1 Sistema distribuido.....	19
M.4.2 Frontend	19
M.4.3 Backend.....	20
M.4.4 Framework	20
M.4.5 Bibliotecas.....	20
M.4.6 Base de datos: Firebase	21
M.5 TÉCNICAS Y HERRAMIENTAS.....	22
M.5.1 Frontend	22
M.5.1.1 HTML.....	22
M.5.1.2 JavaScript	22
M.5.1.3 CSS	22
M.5.2 Framework	23
M.5.2.1 Vue JS.....	23
M.5.2.3 Vuetify.....	23
M.5.2.4 Ionic.....	23
M.5.3 Backend.....	23
M.5.3.1 Node	23

M.5.4 Herramientas	24
M.5.4.1 Visual Studio Code.....	24
M.5.4.2 Visual Paradigm	24
M.5.4.3 EZEstimate	24
M.5.4.4 JSDoc.....	24
M.5.4.5 Microsoft Project.....	25
M.5.4.6 UML	25
M.5.4.7 Npm.....	25
M.6 ASPECTOS RELEVANTES.....	26
M.6.1 Marco de trabajo	26
M.6.2 Estimación de coste y planificación temporal.....	29
M.6.2.1 Estimación de coste y esfuerzo.....	30
M.6.2.2 Planificación temporal.....	31
M.6.3 Especificación de requisitos:.....	37
M.6.3.1 Objetivos del sistema.....	37
M.6.3.2 Requisitos de información.....	39
M.6.3.3 Requisitos no funcionales.....	40
M.6.3.4 Requisitos funcionales.....	42
M.6.4 Análisis de requisitos	51
M.6.4.1 Modelo de dominio	52
M.6.4.2 Paquete de análisis.....	53
M.6.4.3 Realización de casos de uso	55
M.6.5 Diseño del sistema	56
M.6.5.1 Patrones arquitectónicos.....	56
M.6.5.2 Subsistemas de diseño	57
M.6.5.3 Clases de diseño	58
M.6.5.4 Realización de casos de uso	59
M.6.5.5 Diseño de la base de datos.....	61
M.6.5.6 Modelo de despliegue.....	62
M.6.6 Implementación.....	63
M.6.7 Pruebas	64
M.6.8 Documentación	65
M.6.9 Funcionalidad del sistema.....	66
M.6.9.1 Autenticación.....	66
M.6.9.2 Gestión de ejercicios	68

M.6.9.3 Gestión de comidas	71
M.6.9.4 Gestión de usuario	73
M.7 CONCLUSIONES	75
M.8 LINEAS FUTURAS	76
M.9.REFERENCIAS	77

INDICE DE TABLAS

Tabla 1: Comparativa de estimaciones.....	36
Tabla 2: OBJ-Gestión de comidas.....	38
Tabla 3: OBJ-Gestión de ejercicios.....	38
Tabla 4: IRQ-Información comidas	39
Tabla 5: IRQ-Información ejercicios	40
Tabla 6: NFR-Usabilidad	41
Tabla 7:UC-Mostrar comidas.....	45
Tabla 8: UC-Añadir comida.....	46
Tabla 9: UC-Eliminar comida.....	47
Tabla 10: UC-Añadir comida a favoritos.....	48
Tabla 11: Eliminar comida de favoritos.....	49
Tabla 12: UC- Eliminar comida de favoritos.....	50

ÍNDICE DE ILUSTRACIONES

Ilustración 1: 8fit	15
Ilustración 2: MyFitnessPal.....	16
Ilustración 3: Sistema distribuido.....	19
Ilustración 4: Firebase	21
Ilustración 5: Diagrama de fases proceso unificado.....	26
Ilustración 6: Fases e iteraciones del proyecto.....	28
Ilustración 7: Tareas de una Iteración del proyecto	29
Ilustración 8: Resultados estimación coste EZEstimate.....	30
Ilustración 9: Horario de trabajo	31
Ilustración 10: Diagrama de Gantt	35
Ilustración 11: Estadísticas planificación temporal.....	36
Ilustración 12: Diagrama de paquetes	42
Ilustración 13: Diagrama de actores.....	43
Ilustración 14: diagrama casos de uso del paquete gestión de comidas	44
Ilustración 15: Modelo de dominio	52
Ilustración 16: Diagrama paquete de análisis.....	53
Ilustración 17: diagrama clases de análisis Gestión Comidas	54
Ilustración 18: Diagrama clases de análisis Gestión Ejercicios	54
Ilustración 19: diagrama de secuencia UC-Añadir comida.....	55
Ilustración 20: Diagrama de secuencia UC-Eliminar comida	55
Ilustración 21: Diagrama patrón MVVM.....	56
Ilustración 22: Diagrama subsistemas de diseño.....	57
Ilustración 23: Diagrama clases de diseño VistaModelo	58
Ilustración 24: Realización casos de uso UC-Añadir comida	59
Ilustración 25: Realización casos de uso UC-Eliminar comida	60
Ilustración 26: Diseño de Firebase	61
Ilustración 27: Diagrama modelo de despliegue	62
Ilustración 28: Interfaz gráfica documentación técnica.....	65
Ilustración 29: Pantalla inicio de sesión.....	66
Ilustración 30: Pantalla restablecimiento contraseña	67
Ilustración 31: Email modificación contraseña.....	67
Ilustración 32: Pantalla de ejercicios.....	68
Ilustración 33: Pantalla añadir ejercicio cronómetro.....	69

Ilustración 34: Pantalla añadir ejercicio manualmente.....	70
Ilustración 35:Pantalla de comidas.....	71
Ilustración 36: Pantalla de Añadir/Eliminar comida	72
Ilustración 37: Pantalla Perfil.....	73
Ilustración 38: Pantalla de cerrar sesión.....	74

M.1 INTRODUCCIÓN

El propósito del proyecto es simplificar la gestión de la actividad física y la alimentación de los usuarios, con el fin de optimizar la organización de sus rutinas y alcanzar objetivos de salud. El enfoque central es capacitar a los usuarios para estructurar su día de manera óptima y así fortalecer sus metas de salud y calidad de vida. A través de esta iniciativa, los usuarios pueden registrar y rastrear sus actividades deportivas, al mismo tiempo que monitorizan las calorías consumidas durante estas sesiones.

En el ámbito alimenticio, los usuarios tienen la capacidad de diseñar una dieta personalizada y organizar su ingesta calórica diaria. Esto se basa en datos técnicos proporcionados por la aplicación, facilitando así el equilibrio entre la energía consumida y gastada. La integración de estos dos aspectos en una única aplicación la hace especialmente atractiva para los usuarios, ya que la mayoría de las herramientas disponibles se enfocan únicamente en uno de estos aspectos.

Además de su enfoque integral en alimentación y actividad física, es importante resaltar que esta aplicación se distingue por su interfaz intuitiva, lo que permite a los usuarios gestionar la información resultante de sus esfuerzos mediante gráficos y datos visuales de fácil comprensión. El objetivo fundamental es convertir los datos diarios en información valiosa, proporcionando una herramienta que sea sencilla de entender y utilizar.

Esta herramienta motiva a los usuarios a conseguir sus objetivos de bienestar de manera efectiva y placentera. Lo que la diferencia de otras soluciones en el mercado es su capacidad para sobresalir en un entorno altamente competitivo, donde la salud y el bienestar son cada vez más prominentes en todo el mundo. Ya sea a través de numerosas aplicaciones especializadas o profesionales que ganan visibilidad en las redes sociales, el campo de la salud y el bienestar está en constante crecimiento.

La ejecución del proyecto se concretará mediante la implementación de una aplicación móvil, que asumirá el papel de la interfaz primordial destinada a los usuarios. Para llevar a cabo este proceso de desarrollo, se hará uso del framework VueJS de JavaScript, reconocido por su capacidad para edificar interfaces dinámicas sobre una base estructural sólida y altamente eficaz.

Como un componente adicional y crucial en la arquitectura, se integrará una base de datos que se encargará de gestionar los datos vitales del sistema. En este sentido, se empleará Firebase, una plataforma ampliamente reconocida por su habilidad para almacenar y administrar datos de manera efectiva. Firebase asumirá la responsabilidad de manejar los registros relacionados con los usuarios, así como los históricos correspondientes a los ejercicios realizados y los alimentos consumidos. De esta manera, la aplicación garantizará una experiencia fluida y completa para los usuarios al disponer de una interfaz intuitiva respaldada por una sólida infraestructura tecnológica.

A lo largo de este informe se analizarán en detalle los elementos más destacados que conforman el proceso de desarrollo integral del proyecto, englobando los siguientes puntos clave:

- **Análisis comparativo con aplicaciones afines:** se llevará a cabo una exhaustiva comparación con otras aplicaciones similares presentes en el mercado. Esta evaluación permitirá identificar fortalezas, debilidades y oportunidades, proporcionando un contexto esencial para la diferenciación del proyecto.
- **Objetivos fundamentales:** se expondrán con claridad los objetivos primordiales que rigen este proyecto. Estos objetivos actuarán como el marco orientador para todas las

etapas de desarrollo, estableciendo metas medibles y direccionando los esfuerzos hacia la consecución exitosa de los resultados esperados.

- **Fundamentos teóricos esenciales:** se introducirán los conceptos teóricos esenciales que constituyen los cimientos del proyecto. Estos conceptos brindarán el conocimiento necesario para una comprensión profunda y adecuada de las técnicas, metodologías y soluciones implementadas a lo largo del proceso.
- **Técnicas y herramientas utilizadas:** se detallarán las técnicas y herramientas específicas empleadas para desarrollar el proyecto. Desde metodologías de diseño hasta plataformas de desarrollo, se describirá en qué medida cada una de estas herramientas contribuyó al avance y éxito del proyecto.
- **Aspectos críticos:** se analizarán los factores críticos que influyeron en el desarrollo y la ejecución del proyecto. Esto incluirá desafíos superados, decisiones clave tomadas durante el proceso y cualquier otro elemento que haya tenido un impacto significativo en la dirección y el resultado del proyecto.
- **Conclusiones:** se presentarán las conclusiones extraídas de la implementación del proyecto. Estas conclusiones evaluarán en qué medida se cumplieron los objetivos iniciales, qué lecciones se aprendieron durante el proceso y cómo las soluciones implementadas abordan los problemas planteados.
- **Perspectivas futuras:** se explorarán las posibles direcciones futuras que podrían derivarse a partir del proyecto completado. Estas perspectivas podrían abarcar áreas de mejora, expansiones o funcionalidades adicionales que podrían enriquecer y ampliar el funcionamiento de la aplicación.

Para obtener información más exhaustiva, en esta memoria se contempla seis anexos que profundizan en los detalles:

- **Anexo I** - Plan de proyecto.
- **Anexo II** - Especificación de requisitos.
- **Anexo III** - Análisis de requisitos.
- **Anexo IV** - Diseño del sistema software.
- **Anexo V** - Documentación Técnica.
- **Anexo VI** - Manual de usuario.

M.2

M.2.1 8fit

8fit es una aplicación de fitness y nutrición que ofrece planes de entrenamiento y comidas personalizadas. Proporciona rutinas de ejercicios en casa, recetas saludables y fáciles de preparar, y herramientas de seguimiento para monitorizar el progreso del usuario hacia sus objetivos de salud y bienestar.

La diferenciación clave entre mi aplicación y esta alternativa radica en las funcionalidades de pago. Mientras que mi aplicación ofrece una experiencia completa sin costo alguno, la competencia requiere una suscripción de pago para acceder a características vitales como la iniciación de actividades deportivas o el registro detallado de la alimentación.

A continuación, se va a mostrar cómo se ve la aplicación:

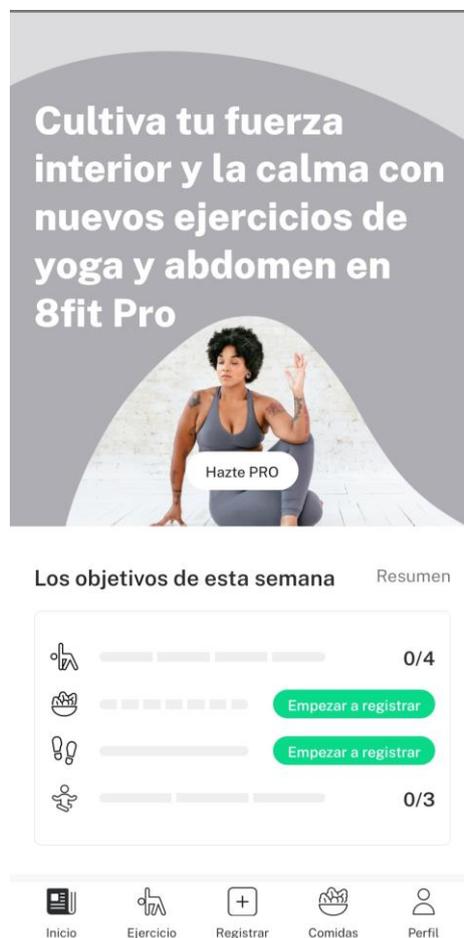


ILUSTRACIÓN 1: 8FIT

M.2.2 MyFitnessPal

MyFitnessPal es una aplicación de salud y bienestar que permite a los usuarios rastrear su ingesta de alimentos y actividad física. Cuenta con una amplia base de datos de alimentos, puede sincronizarse con otras aplicaciones de fitness y ofrece informes personalizados para ayudar a los usuarios a alcanzar sus objetivos de salud.

Aunque esta aplicación es más completa en su versión gratuita que la anterior, presenta una gran desventaja en comparación con la mía. Su interfaz es menos visual y, por ende, menos intuitiva, al no separar claramente el apartado de ejercicios del de alimentación, lo que dificulta la búsqueda de información para el usuario. Adicionalmente, la inclusión de anuncios en la aplicación distrae la atención de los elementos fundamentales de la misma. Contrariamente, mi aplicación ofrece una ventaja al brindar la funcionalidad de organizar las calorías diarias, lo que la hace mucho más personalizada y adaptable a las necesidades individuales de cada usuario.

A continuación, se va a mostrar cómo se ve la aplicación:

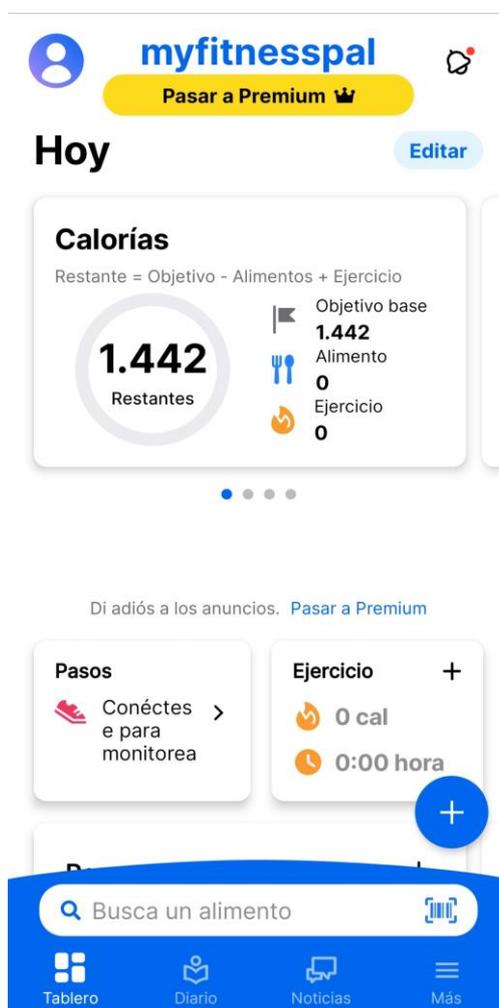


ILUSTRACIÓN 2: MYFITNESSPAL

M.3 OBJETIVOS

En esta sección se detallarán los diferentes propósitos que el proyecto busca alcanzar. Estos objetivos se dividen en dos categorías principales: los del sistema y los personales del desarrollador.

M.3.1 Objetivos del sistema

Para cumplir de manera efectiva con el propósito central del proyecto, se han establecido varios objetivos que deberán alcanzarse:

- **Gestión de autenticación:** el sistema está diseñado para permitir el registro de nuevos usuarios y, para aquellos que ya dispongan de una cuenta, proporciona la funcionalidad para iniciar sesión o cerrar sesión. Además, incorpora una función de recuperación y modificación de la contraseña, brindándole al usuario la posibilidad de cambiar su contraseña en caso de que la haya olvidado.
- **Gestión usuarios:** el sistema administrará meticulosamente la información de los usuarios. Brindará a cada usuario la capacidad de visualizar sus propios datos con claridad y flexibilidad para realizar modificaciones.
- **Gestión de comidas:** el sistema facilitará la visualización, adición y modificación de las diversas opciones alimenticias que se incorporan en el plan nutricional personalizado del usuario.
- **Gestión de ejercicios:** el sistema facilitará la visualización, incorporación y ajuste de los diversos ejercicios que conforman el plan de entrenamiento personalizado del usuario.
- **Gestión de calorías:** el sistema posibilitará la visualización y la modificación de las calorías ingeridas y quemadas por el usuario.
- **Gestión de progresos:** el sistema habilitará funciones para visualizar, incorporar y ajustar tanto las actividades deportivas como los hábitos nutricionales del usuario.
- **Gestión de estadísticas:** el sistema facilitará la visualización de un resumen integrado de la actividad deportiva y nutricional del usuario.

El sistema deberá llevar a cabo unos objetivos no funcionales relacionados con el funcionamiento del sistema. Estos no se centran en lo que hace el sistema, sino en cómo lo hace. De manera eficiente, segura, fácil de usar, compatible con otros sistemas y capaz de manejar un crecimiento futuro. Entre ellos encontramos:

- **Multiplataforma:** el sistema debe ser versátil, con capacidad de funcionar en múltiples plataformas, incluyendo Windows, Linux, Android y más. Debería permitir su uso en una amplia variedad de dispositivos, con un diseño adaptativo que respeta y aprovecha las características específicas de cada plataforma, garantizando una funcionalidad óptima y una experiencia de usuario uniforme y gratificante en todas ellas.
- **Usabilidad:** el sistema debe ser intuitivo y fácil de usar, garantizando que los usuarios puedan navegar de manera eficaz, eficiente y sin complicaciones, permitiendo una experiencia de usuario amigable y satisfactoria.

- **Concurrencia:** el sistema deberá estar diseñado para ser utilizado simultáneamente por múltiples usuarios desde diversos equipos, manteniendo un rendimiento óptimo y garantizando una experiencia de usuario fluida y eficiente.
- **Funcionamiento en tiempo real:** el sistema debe asegurar no sólo la recepción inmediata de datos, sino también su representación eficaz y sin demoras, manteniendo así una actualización constante y en tiempo real para optimizar la interacción y toma de decisiones del usuario.
- **Seguridad de datos:** Es imperativo que el sistema garantice no solo la privacidad, sino también la integridad y la confidencialidad de los datos almacenados. Asimismo, debe implementar rigurosas medidas de control para prevenir cualquier acceso no autorizado, protegiendo así la información confidencial de posibles brechas de seguridad.

Además, el sistema deberá llevar a cabo ciertos objetivos funcionales que son los elementos cruciales que especifican detalladamente las operaciones y actividades que un sistema o sus componentes deben realizar para satisfacer las necesidades de los usuarios y alcanzar los objetivos planteados, alcanzando el correcto funcionamiento del sistema.

M.3.2 Objetivos personales

El objetivo principal de este proyecto es aplicar los conocimientos adquiridos durante la carrera de Grado en Ingeniería Informática mediante la creación de un proyecto completo. En este proyecto, se han utilizado conocimientos de diversas asignaturas de manera integrada y según las necesidades específicas de cada etapa del desarrollo. Además, se ha buscado aprender nuevas habilidades en el ámbito de desarrollo de aplicaciones móviles, como el manejo de Firebase y Vue.

La motivación para adentrarse en nuevas herramientas, como Vue, radica en su alta demanda en el mercado laboral. Adquirir habilidades en estas tecnologías no sólo enriquece el conjunto de competencias, sino que también puede ser un factor determinante para el éxito en futuras oportunidades profesionales. Además de las herramientas, el entendimiento de cómo se construye un proyecto desde cero es fundamental en el mercado laboral actual.

En cuanto al tema que he elegido, la relación entre la alimentación y el deporte, es un asunto que me ha motivado de manera especial a lo largo de mi vida. Siempre he procurado llevar un estilo de vida lo más saludable posible, combinando la práctica deportiva con una alimentación equilibrada. La oportunidad de crear una aplicación personalizada que me permitiera llevar un seguimiento diario ha sido una experiencia extremadamente gratificante. Además, me he sentido inspirado por personas cercanas a mí que han expresado en repetidas ocasiones su deseo de contar con una herramienta similar a la que he desarrollado, para ayudarles en su día a día.

M.4 CONCEPTOS TEÓRICOS

En esta sección se abordarán diversos conceptos que requieren definiciones detalladas para lograr una comprensión completa del proyecto.

M.4.1 Sistema distribuido

En un sistema distribuido encontramos el hardware o el software en diferentes dispositivos interconectados mediante una red y se comunican intercambiando mensajes. El objetivo principal de un sistema distribuido es la cooperación entre estas entidades separadas para lograr un servicio en común. Esto permite compartir recursos, como capacidad de procesamiento y almacenamiento, y brinda mayor redundancia y escalabilidad en comparación con los sistemas centralizados.

La siguiente ilustración representa gráficamente su estructura:

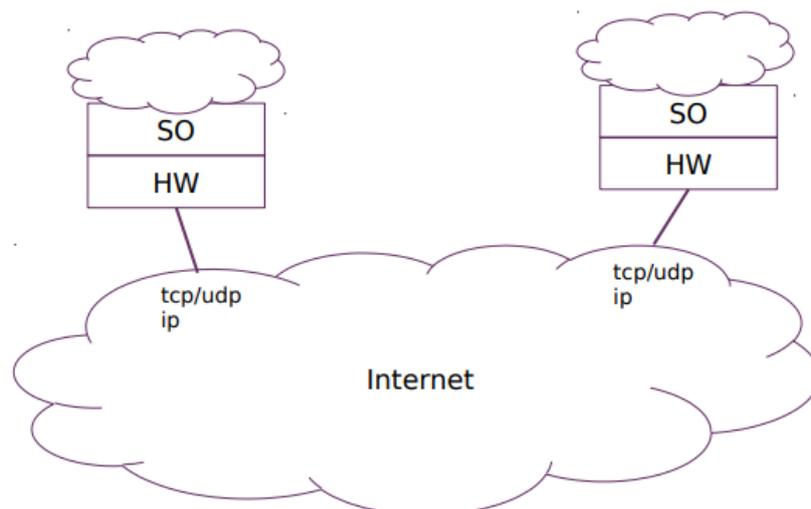


ILUSTRACIÓN 3: SISTEMA DISTRIBUIDO

M.4.2 Frontend

El Frontend es la parte visible y accesible de una aplicación móvil o web, con la que el usuario interactúa directamente. Se encuentra en el lado del cliente y se encarga de presentar la interfaz de usuario, así como de manejar la interacción y la presentación de datos.

Algunos de los lenguajes de programación utilizados en el frontend son HTML, JavaScript y CSS.

M.4.3 Backend

El backend es la otra parte de una aplicación que se encuentra en el lado del servidor y es responsable de gestionar los datos y la comunicación con la base de datos. Los componentes del backend procesan las solicitudes del frontend y proporcionan los datos necesarios para su visualización. Esta capa es invisible para los usuarios y generalmente se encarga de manejar la seguridad, el almacenamiento y la gestión de datos.

Lenguajes como Node.js son comunes en el desarrollo del backend, y es el que ha sido utilizado en este proyecto.

M.4.4 Framework

Un marco de trabajo (framework) es una estructura predefinida que ofrece un conjunto de herramientas, librerías y pautas para desarrollar software de manera más eficiente y coherente. Actúa como una base sobre la cual se construyen aplicaciones, facilitando tareas comunes y brindando una estructura organizada. Los frameworks permiten a los desarrolladores evitar construir todo desde cero y, en cambio, aprovechar soluciones probadas y estandarizadas para componentes como manejo de datos, interfaz de usuario y más.

Los framework utilizados son Vue.js e Ionic. Vue.js se centra en la construcción de interfaces de usuario reactivas para aplicaciones web, mientras que Ionic se basa en el desarrollo de aplicaciones móviles y web utilizando tecnologías web estándar y componentes preconstruidos.

M.4.5 Bibliotecas

Las bibliotecas son conjuntos de código reutilizable y funciones predefinidas que facilitan el desarrollo de software al proporcionar funcionalidades específicas. A diferencia de los frameworks, las bibliotecas no establecen una estructura rígida para el proyecto, sino que permiten a los desarrolladores utilizar sus componentes según sea necesario en sus aplicaciones.

Las bibliotecas utilizadas han sido especialmente para iconos, como son los botones, han desempeñado un papel fundamental en la creación de interfaces visuales atractivas y funcionales.

M.4.6 Base de datos: Firebase

La base de datos utilizada ha sido Firebase. Una plataforma de desarrollo de diferentes plataformas como en nuestro caso que es Android. Creada por Google que proporciona una variedad de funcionalidades y servicios con el fin de ayudar a los desarrolladores a construir y mejorar sus aplicaciones.

Algunos de sus servicios son:

- **Firestore Authentication:** ofrece una solución integral de autenticación de usuarios que abarca la autenticación mediante correo electrónico y contraseña, así como la autenticación a través de proveedores como Google o Facebook.
- **Firestore:** es una base de datos NoSQL más moderna y potente que brinda escalabilidad y capacidades de consulta en tiempo real. Es idónea para aplicaciones web y móviles que demandan una estructura de datos más avanzada.
- **Realtime Database:** proporciona una base de datos NoSQL en tiempo real que permite a las aplicaciones sincronizar datos en tiempo real entre clientes y servidores.
- **Cloud Storage:** facilita el almacenamiento de archivos en la nube, resultando especialmente útil para la gestión eficiente de imágenes, videos y otros recursos multimedia.
- **Analytics:** posibilita el seguimiento del comportamiento de los usuarios en la aplicación para respaldar la toma de decisiones informadas en desarrollo y marketing.

Una de las principales ventajas de Firebase es que muchos de sus servicios funcionan en tiempo real, lo que significa que los cambios en la base de datos se pueden reflejar instantáneamente en la interfaz de usuario de una aplicación. Además, Firebase ofrece una plataforma unificada que permite a los desarrolladores centrarse más en la creación de grandes experiencias de usuario y menos en la gestión de la infraestructura.



ILUSTRACIÓN 4: FIREBASE

M.5 TÉCNICAS Y HERRAMIENTAS

Se procede a detallar las diversas técnicas, herramientas, bibliotecas y frameworks utilizados en la elaboración del proyecto. Estos elementos serán segmentados en cuatro secciones principales, aquellas empleadas en la construcción del frontend, en el framework, en el backend y en las herramientas auxiliares.

M.5.1 Frontend

M.5.1.1 HTML

HTML es un lenguaje de marcas de hipertexto, para elaborar páginas web y aplicaciones móviles. Utiliza etiquetas para identificar contenido como texto, imágenes y videos. Las etiquetas estructuran y dan significado al contenido. Los navegadores interpretan el HTML para mostrar el contenido según las etiquetas y atributos. Se combina con CSS para estilo y JavaScript para interactividad. Es fundamental para una experiencia web efectiva.

M.5.1.2 JavaScript

JavaScript es un popular lenguaje de programación orientado a objetos y basado en prototipos. Se utiliza en páginas web y aplicaciones móviles para mejorar la interacción del usuario. A diferencia de Java, JavaScript se ejecuta en el navegador y se enfoca en la dinámica del usuario. Frameworks como Vue.js e Ionic aprovechan sus capacidades para crear interfaces interactivas y aplicaciones móviles.

M.5.1.3 CSS

CSS es un lenguaje utilizado junto con HTML para gestionar el diseño y la apariencia de aplicaciones web y móviles. Permite definir estilos visuales como colores, tamaños y disposición de elementos. La característica clave es su capacidad para heredar y aplicar estilos de manera jerárquica, lo que ahorra tiempo y facilita la coherencia en el diseño en múltiples páginas. Al separar la presentación del contenido, CSS ofrece flexibilidad y control en la creación de interfaces visuales atractivas y adaptables.

M.5.2 Framework

M.5.2.1 Vue JS

Vue es un marco de JavaScript para crear interfaces de usuario dinámicas. Se basa en HTML, CSS y JavaScript, utiliza componentes reutilizables y un enfoque declarativo para programar. Es popular debido a su facilidad de aprendizaje y a la abundante documentación y tutoriales disponibles en línea. Además, se integra bien con Firebase, la base de datos utilizada.

M.5.2.3 Vuetify

Vuetify es un marco de trabajo desarrollado sobre VueJS, diseñado para agiliza la creación de aplicaciones visualmente atractivas. Ofrece una amplia gama de componentes predefinidos basados en Material Design, lo que permite a los desarrolladores construir interfaces de usuario modernas de manera eficiente.

M.5.2.4 Ionic

Ionic es una herramienta de código abierto que facilita la creación de aplicaciones móviles multiplataforma usando tecnologías como HTML, CSS y JavaScript. Permite desarrollar aplicaciones de alta calidad como Android con una única base de código. Además, se integra con Vue.js para desarrollar interfaces de usuario. Ofrece componentes preconstruidos, simplificando el diseño, y destaca por su flexibilidad y adaptabilidad.

M.5.3 Backend

M.5.3.1 Node

Node.js es un entorno de ejecución de código abierto basado en JavaScript que se centra en operaciones asíncronas y eventos. Diseñado para aplicaciones de red escalables. Su arquitectura eficiente y su capacidad para unificar el desarrollo en JavaScript lo hacen ideal para aplicaciones en tiempo real. Esta herramienta se ha utilizado para ejecutar el código desarrollado en el lado del servidor y que mediante un navegador se pueda visualizar en el lado del cliente. Ofrece una amplia comunidad, gestión de paquetes a través de npm y una variedad de módulos para simplificar el desarrollo.

M.5.4 Herramientas

M.5.4.1 Visual Studio Code

Visual Studio Code es un editor de código gratuito y versátil ampliamente utilizado en programación. Ofrece características como el autocompletado inteligente (IntelliSense) basado en el contexto del código, depuración directa desde el editor y la capacidad de personalización mediante extensiones. Es compatible con tecnologías como Vue y Node, lo que lo hace útil para proyectos que emplean estas tecnologías, como es este proyecto.

M.5.4.2 Visual Paradigm

Visual Paradigm es una herramienta CASE esencial en el desarrollo de software. Permite crear diversos diagramas usando el estándar UML. UML es un lenguaje visual estándar para representar la estructura y comportamiento de sistemas. Su aplicación en los anexos de este proyecto se debe por contar con una experiencia previa, ya que ha sido empleada en algunas de las asignaturas del grado como son Ingeniería del software I e Ingeniería del software II.

M.5.4.3 EZEstimate

EZEstimate es una herramienta que agiliza la estimación de costos y esfuerzos en proyectos de software utilizando la metodología de puntos de casos de uso. Esta metodología cuantifica las funcionalidades del sistema y asigna puntos de complejidad para estimar el trabajo necesario. EZEstimate automatiza este proceso, mejorando la precisión y transparencia de las estimaciones. Se ha empleado en el *Anexo I – Plan de proyecto software*, demostrando su eficacia en la gestión de proyectos de software. Una ventaja adicional radica en su familiaridad, al haber sido empleado a lo largo de la carrera.

M.5.4.4 JSDoc

JSDoc es una sintaxis que posibilita la incorporación de documentación del código fuente de JavaScript. Esto se logra de forma automática a través de comentarios agregados por los desarrolladores en su código, los cuales describen el propósito, el comportamiento y los tipos de datos de las variables, funciones y clases en JavaScript. Esta práctica simplifica enormemente la generación de documentación del código, ya que produce automáticamente una página HTML que contiene toda la información documentada.

M.5.4.5 Microsoft Project

Microsoft Project es una aplicación de software desarrollada por Microsoft diseñada para la efectiva gestión de proyectos. Esta herramienta permite la creación de planes detallados que incluyen la definición de tareas, duraciones, los recursos necesarios y las interdependencias entre ellas. Facilita la asignación eficiente de recursos humanos y materiales a cada tarea, garantizando una gestión óptima de los recursos disponibles. Además, ofrece la ventaja de realizar un seguimiento en tiempo real del progreso del proyecto, lo que simplifica la identificación de posibles desviaciones y la adopción de medidas correctivas oportunas. También brinda la posibilidad de establecer fechas límite para asegurar que el proyecto se desarrolle de manera eficiente y dentro de los plazos establecidos.

En particular, Microsoft Project ha sido empleado en la elaboración del *Anexo I – Plan de proyecto software*, aprovechando su familiaridad debido a su empleo en la asignatura de Gestión de Proyectos.

M.5.4.6 UML

El Unified Modeling Language (UML) destaca como el lenguaje de modelado de sistemas de software más ampliamente reconocido y empleado en la actualidad. Su función principal radica en la visualización, construcción y documentación de sistemas. UML constituye un lenguaje de modelado diseñado para la representación visual, la especificación precisa, la construcción efectiva y la documentación exhaustiva de diversas partes que componen un sistema de software. Este lenguaje es versátil, adaptándose sin dificultades a distintos procesos de desarrollo a lo largo de todo el ciclo de vida del proyecto, y puede aplicarse con éxito en diversos dominios de aplicación y plataformas de implementación. Se cuenta con experiencia previa durante el grado, UML se ha empleado en asignaturas como Ingeniería del Software I e Ingeniería del Software II. Estas instancias han permitido familiarizarnos con las ventajas y aplicaciones prácticas de UML.

M.5.4.7 Npm

El Administrador de Paquetes de Node, conocido como NPM, es la herramienta principal para gestionar paquetes en Node.js, un entorno de ejecución destinado al uso de JavaScript. Este utilitario se opera a través de la terminal y brinda la capacidad de instalar diversas bibliotecas y herramientas de manera sencilla y eficiente.

M.6 ASPECTOS RELEVANTES

En este apartado, se brindará una descripción minuciosa de los elementos y componentes más significativos durante la ejecución del proyecto. Para obtener una comprensión más exhaustiva sobre estos aspectos, se proporcionará información adicional en los anexos correspondientes.

M.6.1 Marco de trabajo

En el desarrollo del proyecto se ha utilizado el marco de trabajo del Proceso Unificado. Este es un enfoque versátil que permite adaptarse a diversos tipos de sistemas de software. Esta elección se basa en su naturaleza iterativa e incremental, que proporciona una notable flexibilidad en la gestión del proyecto.

El Proceso Unificado se estructura en cuatro fases fundamentales: Inicio, Elaboración, Construcción y Transición. Además, el Proceso Unificado se descompone en diversas disciplinas especializadas que permiten un abordaje minucioso y organizado. Estas disciplinas incluyen Modelado del Negocio, Definición de Requisitos, Análisis, Diseño, Implementación, Pruebas, Configuración y Gestión del Proyecto. Cada disciplina se centra en un aspecto particular del proceso, contribuyendo a la cohesión y efectividad del mismo. Estos detalles son visibles en la representación gráfica que se presenta a continuación.

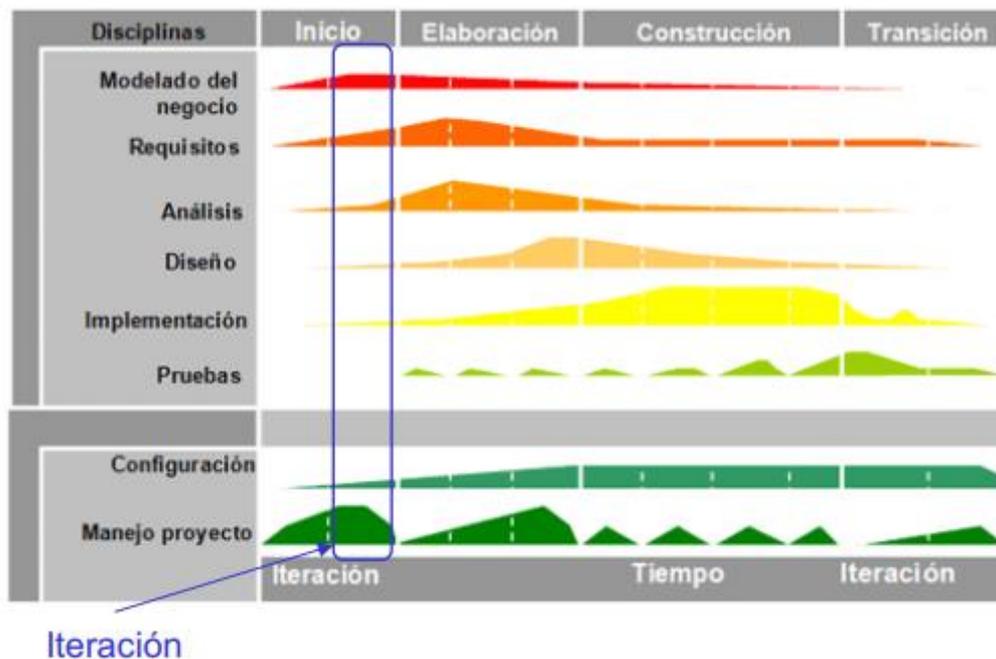


ILUSTRACIÓN 5: DIAGRAMA DE FASES PROCESO UNIFICADO

- **Inicio:** en esta fase se establece el alcance inicial del proyecto y se identifican los objetivos, los requisitos principales y las restricciones.
- **Elaboración:** en esta fase, se definen y refinan los requisitos de manera más detallada, se establece la arquitectura del sistema y se desarrolla un prototipo funcional. Se realizan análisis de riesgos y se crea una base sólida para el desarrollo continuo. Al final de esta fase, se revisa y valida la arquitectura para asegurar su viabilidad.
- **Construcción:** durante esta fase, se desarrolla el software real a partir de los diseños y las especificaciones creadas en las fases anteriores. Se llevan a cabo iteraciones de desarrollo, se implementan nuevas funcionalidades y se realizan pruebas constantes para garantizar la calidad del código. Esta fase se centra en la producción y la construcción del producto.
- **Transición:** en esta, el software se prepara para su lanzamiento y se realiza la transición de su desarrollo a la operación y el uso del cliente. Se realizan pruebas finales, se corrigen los errores restantes y se prepara la documentación necesaria. Una vez que el software cumple con los criterios de aceptación, se procede a su implementación y despliegue.

Es importante mencionar que el Proceso Unificado es iterativo e incremental, lo que significa que las fases no son rígidas y lineales, sino que se pueden repetir y ajustar a medida que se obtiene un mayor entendimiento del proyecto y sus requerimientos.

Las fases y las iteraciones del proyecto, tal como se han definido en Microsoft Project, se presentan a continuación:

▣ FASE DE INICIO	8,5 días	lun 10/10/22	jue 20/10/22
▷ Iteración 1	4,5 días	lun 10/10/22	vie 14/10/22
Hito iteración 1	0 días	vie 14/10/22	vie 14/10/22
▷ Iteración 2	4 días	vie 14/10/22	jue 20/10/22
Hito iteración 2	0 días	jue 20/10/22	jue 20/10/22
Hito fase de inicio	0 días	jue 20/10/22	jue 20/10/22
▣ FASE DE ELABORACIÓN	50 días	jue 20/10/22	jue 29/12/22
▷ Iteración 1	11,5 días	jue 20/10/22	vie 04/11/22
Hito iteración 1	0 días	vie 04/11/22	vie 04/11/22
▷ Iteración 2	14 días	vie 04/11/22	jue 24/11/22
Hito iteración 2	0 días	jue 24/11/22	jue 24/11/22
▷ Iteración 3	14 días	jue 24/11/22	mié 14/12/22
Hito iteración 3	0 días	mié 14/12/22	mié 14/12/22
▷ Iteración 4	10,5 días	mié 14/12/22	jue 29/12/22
Hito iteración 4	0 días	jue 29/12/22	jue 29/12/22
Hito fase de elaboración	0 días	jue 29/12/22	jue 29/12/22
▣ FASE DE CONSTRUCCIÓN	78,5 días	jue 29/12/22	jue 20/04/23
▷ Iteración 1	16,5 días	jue 29/12/22	vie 20/01/23
Hito iteración 1	0 días	vie 20/01/23	vie 20/01/23
▷ Iteración 2	21 días	vie 20/01/23	lun 20/02/23
Hito iteración 2	0 días	lun 20/02/23	lun 20/02/23
▷ Iteración 3	25 días	lun 20/02/23	lun 27/03/23
Hito iteración 3	0 días	lun 27/03/23	lun 27/03/23
▷ Iteración 4	16 días	lun 27/03/23	jue 20/04/23
Hito iteración 4	0 días	jue 20/04/23	jue 20/04/23
Hito fase de construcción	0 días	jue 20/04/23	jue 20/04/23
▣ FASE DE TRANSICIÓN	17 días	jue 20/04/23	lun 15/05/23
▷ Iteración 1	11 días	jue 20/04/23	vie 05/05/23
Hito iteración 1	0 días	vie 05/05/23	vie 05/05/23
▷ Iteración 2	6 días	vie 05/05/23	lun 15/05/23
Hito iteración 2	0 días	lun 15/05/23	lun 15/05/23
Hito fase de transición	0 días	lun 15/05/23	lun 15/05/23

ILUSTRACIÓN 6: FASES E ITERACIONES DEL PROYECTO

Además de estas fases, el Proceso Unificado también se basa en disciplinas que agrupan las actividades y tareas en áreas específicas. Estas disciplinas trabajan de manera interconectada a lo largo de las fases para garantizar un enfoque completo y coherente en el desarrollo del software. A continuación, se muestra un ejemplo con la Iteración 1 de la fase de Inicio:

▣ FASE DE INICIO	8,5 días	lun 10/10/22	jue 20/10/22		
▣ Iteración 1	4,5 días	lun 10/10/22	vie 14/10/22		
▣ Modelado de Negocio	1,5 días	lun 10/10/22	mar 11/10/22		
Reunión con cliente	0,5 días	lun 10/10/22	lun 10/10/22		Maria
Reunión con equipo	0,5 días	lun 10/10/22	lun 10/10/22	4	Maria
Análisis DAFO	0,5 días	mar 11/10/22	mar 11/10/22	5	Maria
▣ Requisitos	2,5 días	mié 12/10/22	vie 14/10/22		
Identificar Actores	0,5 días	mié 12/10/22	mié 12/10/22	5	Maria
Identificar y documentar objetivos generales	0,5 días	jue 13/10/22	jue 13/10/22	6	Maria
Identificar y documentar RNF básicos	0,5 días	vie 14/10/22	vie 14/10/22	5	Maria
Hito iteración 1	0 días	vie 14/10/22	vie 14/10/22	2	

ILUSTRACIÓN 7: TAREAS DE UNA ITERACIÓN DEL PROYECTO

En la primera iteración de la fase de Inicio del Proceso Unificado, las tareas se centran en establecer una comprensión sólida del negocio y los requisitos clave del proyecto. Estas tareas son cruciales para definir la dirección del proyecto y asegurarse de que todas las partes involucradas tengan una visión clara de lo que se está buscando lograr.

M.6.2 Estimación de coste y planificación temporal

En esta sección del proyecto, el primer paso consiste en llevar a cabo la estimación de costos y la planificación temporal. Estas actividades nos permiten identificar las distintas tareas que deberán ser realizadas y conseguir una estimación del tiempo requerido para completar cada una de ellas.

Para acceder a información más detallada se recomienda consultar el *Anexo I-Plan de proyecto*.

M.6.2.1 Estimación de coste y esfuerzo

La estimación del coste y esfuerzo en la gestión de proyectos implica determinar la relación entre los recursos humanos involucrados y el tiempo necesario para completar el proyecto, lo que a su vez permite estimar su duración. Para completar esta estimación, se ha empleado una métrica conocida como Puntos de Casos de Uso (UCP) que se utiliza para medir la funcionalidad representada en casos de uso. Esta técnica resulta especialmente apropiada para proyectos que siguen la metodología UML (Lenguaje de Modelado Unificado), como es el caso de este proyecto en particular. El enfoque UCP contribuye a optimizar tanto la eficiencia como la efectividad del proceso de estimación.

La metodología UCP asigna una complejidad específica tanto a los actores involucrados en el sistema como a los casos de uso que describen sus interacciones. Además, se consideran factores ambientales y de complejidad técnica. Estos elementos conforman la base para calcular el esfuerzo requerido en el desarrollo del proyecto. Para realizar dicho cálculo, se utiliza la herramienta EZEstimate, la cual proporciona los siguientes resultados finales:

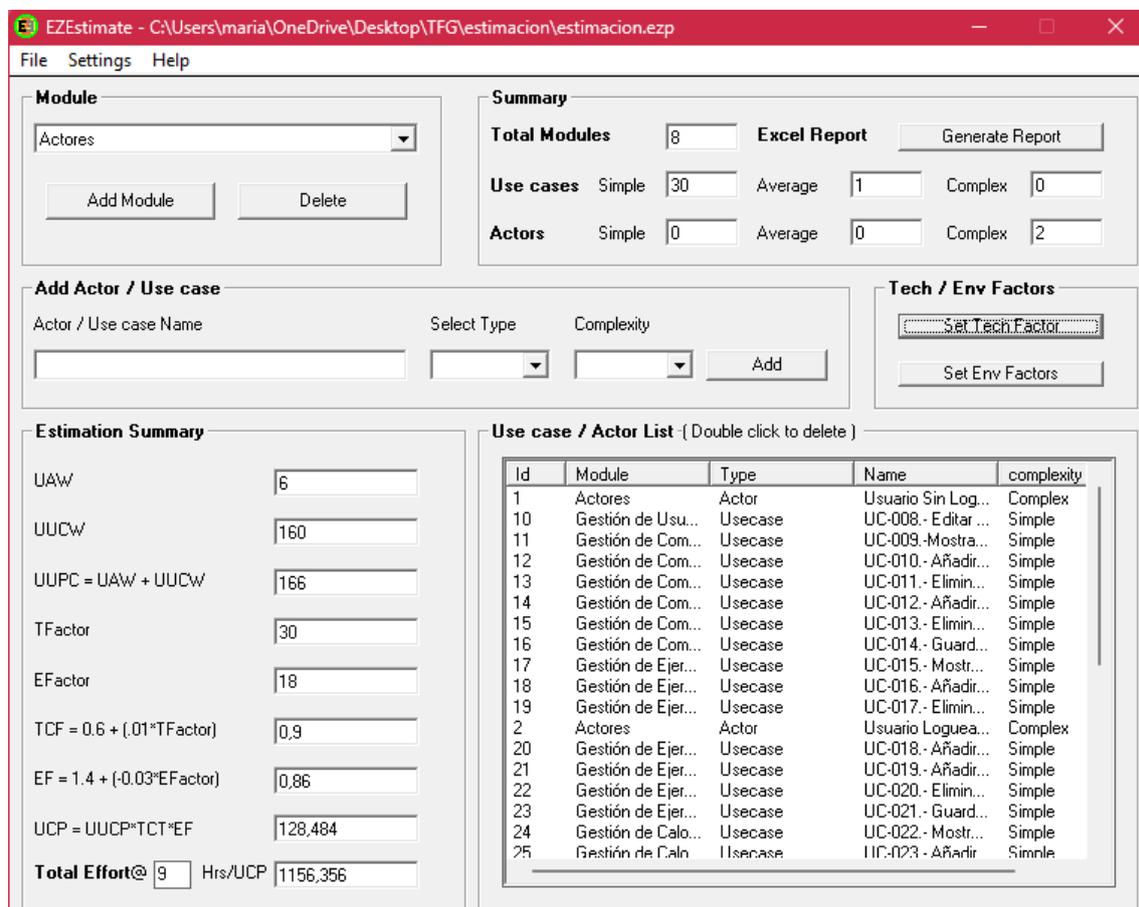


ILUSTRACIÓN 8: RESULTADOS ESTIMACIÓN COSTE EZESTIMATE

Con los datos proporcionados, se calcula una estimación de 1156.35 horas. Si se dedican 8 horas al día, esto equivaldría a un total de 144,54 días necesarios para completar el proyecto por una única persona.

$$1156,35 \text{ horas} / 8 \text{ horas/día} = 144,54 \text{ días}$$

M.6.2.2 Planificación temporal

La planificación temporal desempeña un papel fundamental al posibilitar un análisis minucioso del proyecto. Esta etapa es esencial para descomponer el proyecto en elementos más sencillos, identificando constantemente las tareas ya completadas y proporcionando una estimación precisa de la duración total del proyecto.

Este proceso de planificación se lleva a cabo en la fase inicial del proyecto, un momento estratégico que brinda una comprensión clara de las tareas que se deben abordar en cada etapa. Esto a su vez permite supervisar el cumplimiento de los plazos establecidos y resaltar las tareas críticas que requieren una atención especial. De esta manera, se evitan posibles retrasos en la ejecución del proyecto.

Para poder realizar esta planificación, se hace uso de la herramienta Microsoft Project. Con esta herramienta, se crea un horario de trabajo que se ajusta a las necesidades individuales del desarrollador o equipo, en este caso es de un único desarrollador. Se ha fijado una jornada de 8 horas diarias. El horario comprende dos periodos: de 9:00 a 15:00 y de 16:00 a 18:00, extendiéndose desde el lunes hasta el viernes, como vemos en la siguiente imagen:

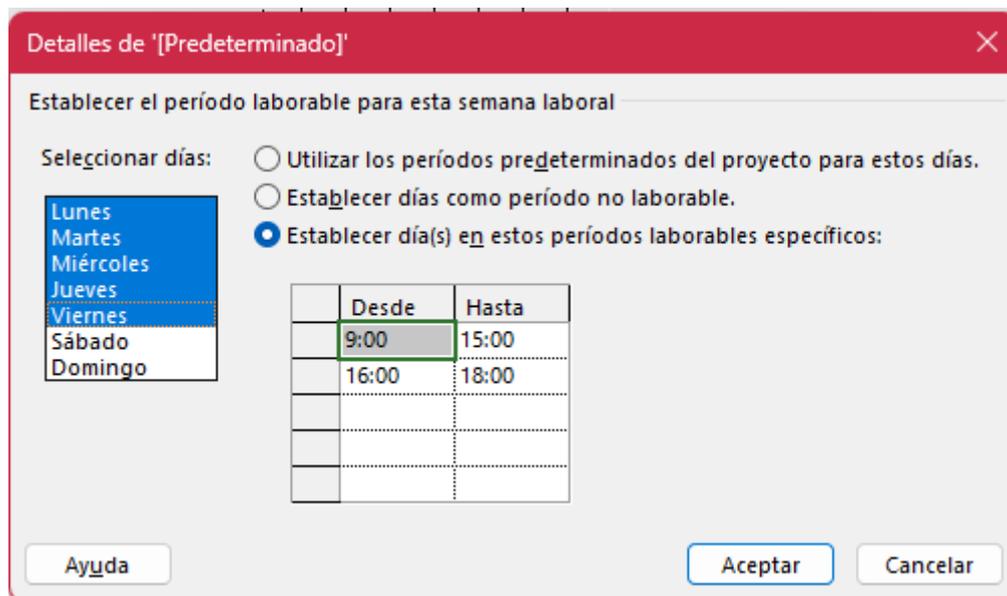
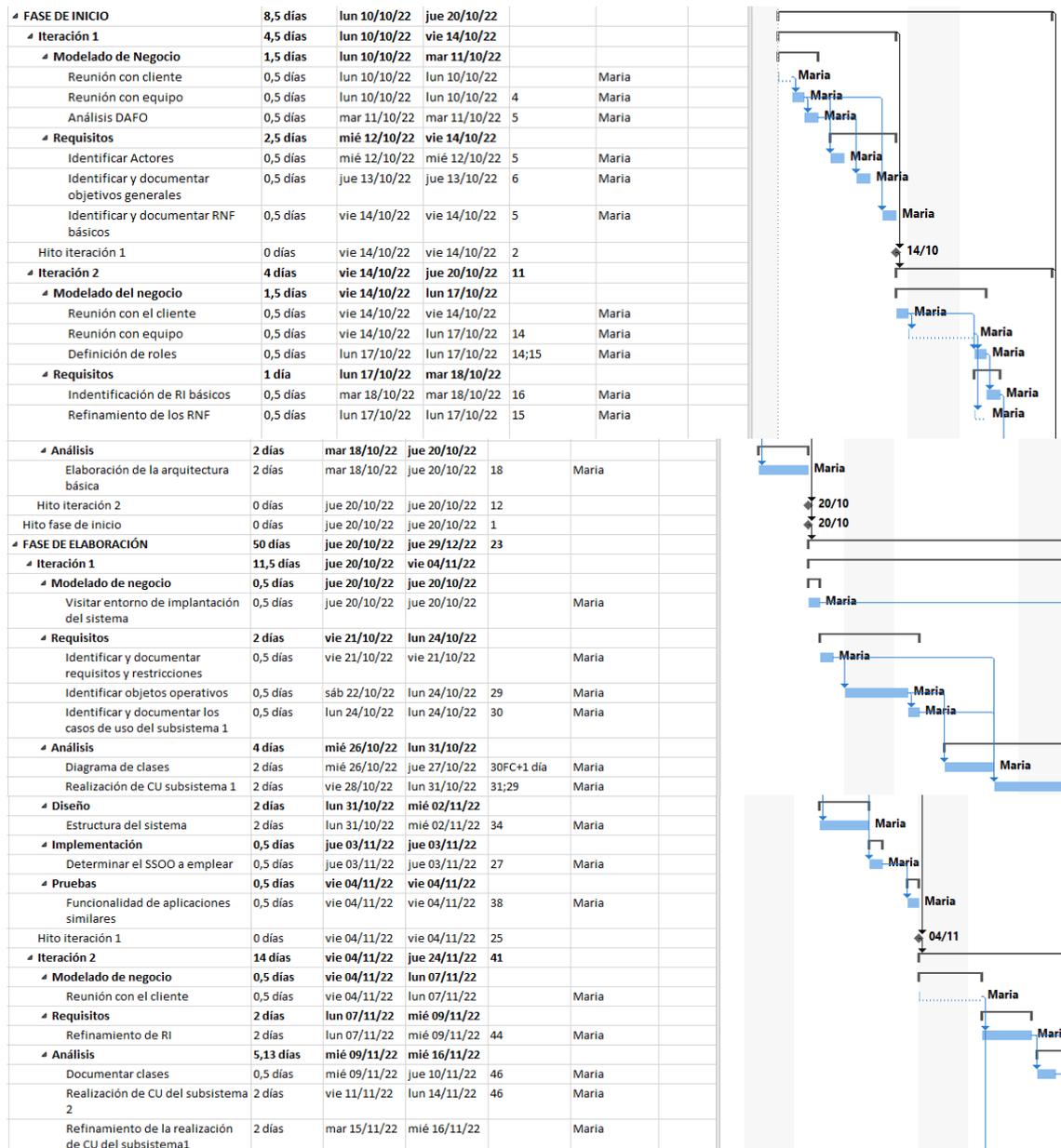


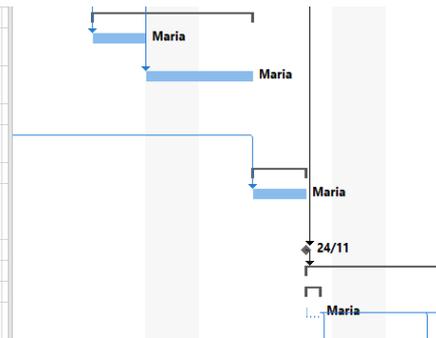
ILUSTRACIÓN 9: HORARIO DE TRABAJO

Al igual que se establece el horario de trabajo, también se definen los días festivos. En este sentido, se han designado aquellos que coinciden con las festividades nacionales y regionales de la comunidad autónoma de Castilla y León.

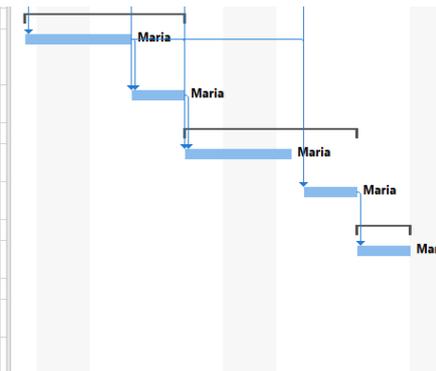
Las diferentes fases y las iteraciones del proyecto se representan de manera visual mediante un diagrama de Gantt. Este diagrama proporciona una representación gráfica de las diversas tareas y su secuencia en el tiempo mediante una barra colocada sobre la escala, cuya longitud refleja la duración de la tarea y cuya posición indica su relación con el punto de inicio del proyecto. Lo que facilita la comprensión de la planificación y la monitorización del progreso del proyecto. Este diagrama se muestra a continuación:



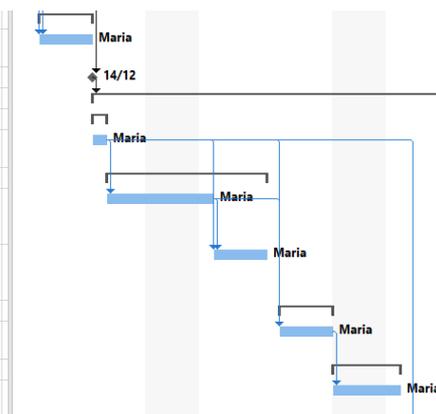
➤ Diseño	4 días	jue 17/11/22	mar 22/11/22		
Refinamiento subsistema de diseño	2 días	jue 17/11/22	vie 18/11/22	49	Maria
Realización de clases de diseño del subsistema 1	2 días	sáb 19/11/22	mar 22/11/22	48;50	Maria
➤ Implementación	0,5 días	lun 07/11/22	lun 07/11/22		
Determinar características huésped del sistema	0,5 días	lun 07/11/22	lun 07/11/22	44	Maria
➤ Pruebas	2 días	mié 23/11/22	jue 24/11/22		
Rendimiento y funcionalidad de aplicaciones similares en entornos similares al huésped	2 días	mié 23/11/22	jue 24/11/22	55	Maria
Hito iteración 2	0 días	jue 24/11/22	jue 24/11/22	42	
➤ Iteración 3	14 días	jue 24/11/22	mié 14/12/22	58	
➤ Modelado del negocio	0,5 días	jue 24/11/22	vie 25/11/22		
Reunión con desarrolladores previa a la construcción	0,5 días	jue 24/11/22	vie 25/11/22		Maria



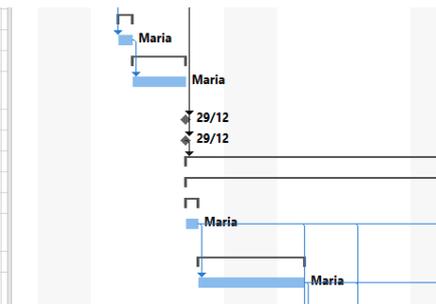
➤ Requisitos	4 días	vie 25/11/22	jue 01/12/22		
Refinamiento de requisitos y restricciones de los objetivos operativos	2 días	vie 25/11/22	mar 29/11/22	61	Maria
Realizar matrices de rastreabilidad (OBJ-REQ)	2 días	mar 29/11/22	jue 01/12/22	61;63	Maria
➤ Análisis	4,5 días	jue 01/12/22	mié 07/12/22		
Refinamiento de la realización de CU del subsistema 2	2 días	jue 01/12/22	lun 05/12/22	61;63;64	Maria
Realización de CU del subsistema 3	2 días	mar 06/12/22	mié 07/12/22	61;63	Maria
➤ Diseño	2 días	mié 07/12/22	vie 09/12/22		
Realización de clases de diseño del subsistema 2	2 días	mié 07/12/22	vie 09/12/22	67	Maria
➤ Implementación	1 día	lun 12/12/22	lun 12/12/22		
Determinar el lenguaje de programación	0,5 días	lun 12/12/22	lun 12/12/22	61	Maria
Determinar lógica de programación	0,5 días	lun 12/12/22	lun 12/12/22	71	Maria



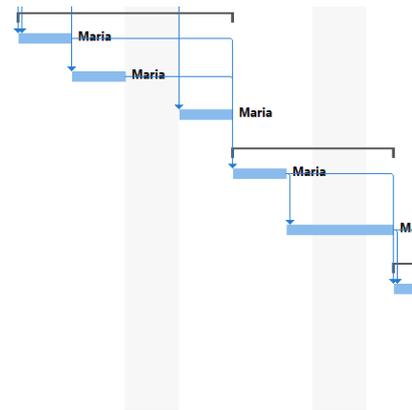
➤ Pruebas	2 días	mar 13/12/22	mié 14/12/22		
Pruebas unitarias del lenguaje de programación	2 días	mar 13/12/22	mié 14/12/22	71;72	Maria
Hito iteración 3	0 días	mié 14/12/22	mié 14/12/22	59	
➤ Iteración 4	10,5 días	mié 14/12/22	jue 29/12/22	75	
➤ Modelo del negocio	0,5 días	mié 14/12/22	jue 15/12/22		
Reunión con desarrolladores previa a la construcción	0,5 días	mié 14/12/22	jue 15/12/22		Maria
➤ Requisitos	4 días	jue 15/12/22	mié 21/12/22		
Refinamiento de requisitos y restricciones de los objetivos operativos	2 días	jue 15/12/22	lun 19/12/22	78	Maria
Realizar matrices de rastreabilidad (REQUISITOS-REQUISITOS)	2 días	lun 19/12/22	mié 21/12/22	78;80	Maria
➤ Análisis	2 días	jue 22/12/22	vie 23/12/22		
Refinamiento de la realización de Casos de Uso subsistema 3	2 días	jue 22/12/22	vie 23/12/22	78;80	Maria
➤ Diseño	0,5 días	vie 23/12/22	lun 26/12/22		
Realización clases diseño subsistema 3	0,5 días	vie 23/12/22	lun 26/12/22	83	Maria



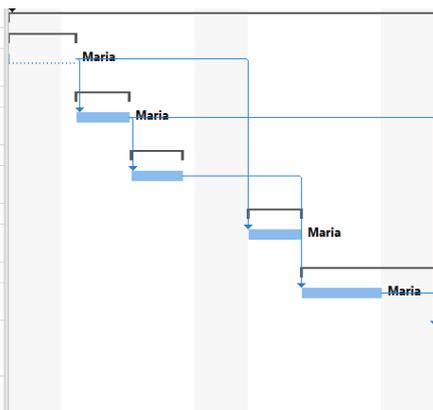
➤ Implementación	0,5 días	mar 27/12/22	mar 27/12/22		
Determinar el sistema de BD	0,5 días	mar 27/12/22	mar 27/12/22	78	Maria
➤ Pruebas	2 días	mar 27/12/22	jue 29/12/22		
Pruebas unitarias de BD en sistema operativo y entorno	2 días	mar 27/12/22	jue 29/12/22	87	Maria
Hito iteración 4	0 días	jue 29/12/22	jue 29/12/22	76	
Hito fase de elaboración	0 días	jue 29/12/22	jue 29/12/22	24	
➤ FASE DE CONSTRUCCIÓN	78,5 días	jue 29/12/22	jue 20/04/23	91	
➤ Iteración 1	16,5 días	jue 29/12/22	vie 20/01/23		
➤ Modelado del negocio	0,5 días	jue 29/12/22	jue 29/12/22		
Reunión con el cliente comprobación requisitos	0,5 días	jue 29/12/22	jue 29/12/22		Maria
➤ Requisitos	2 días	jue 29/12/22	lun 02/01/23		
Validación de todos los requisitos	2 días	jue 29/12/22	lun 02/01/23	95	Maria



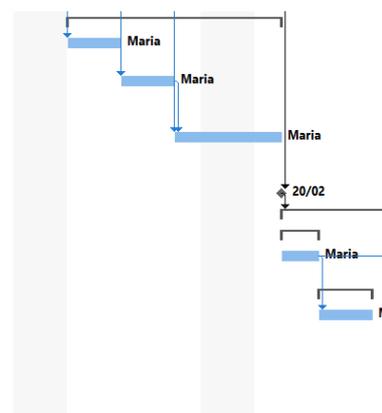
4 Diseño	6 días	lun 02/01/23	mar 10/01/23		
Refinamiento de las clases de diseño del subsistema 1	2 días	lun 02/01/23	mié 04/01/23	95;97	Maria
Refinamiento de las clases de diseño del subsistema 2	2 días	jue 05/01/23	vie 06/01/23	95;97	Maria
Refinamiento de las clases de diseño del subsistema 3	2 días	lun 09/01/23	mar 10/01/23	95;97	Maria
4 Implementación	4 días	mié 11/01/23	lun 16/01/23		
Implementación de los componentes separados del subsistema 1	2 días	mié 11/01/23	jue 12/01/23	99;100	Maria
Implementación de la BD para subsistema 1	2 días	jue 12/01/23	lun 16/01/23	103	Maria
4 Pruebas	4 días	lun 16/01/23	vie 20/01/23		
Pruebas unitarias para cada componente JSON del subsistema	2 días	lun 16/01/23	mié 18/01/23	103;104	Maria
Pruebas unitarias de cada componente del subsistema 1 en SQL	2 días	mié 18/01/23	vie 20/01/23	104;106	Maria
Hito iteración 1	0 días	vie 20/01/23	vie 20/01/23	93	



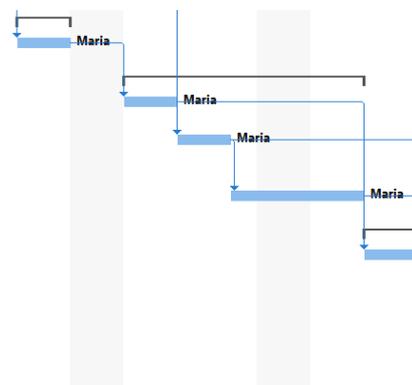
4 Iteración 2	21 días	vie 20/01/23	lun 20/02/23	108	
4 Modelado del negocio	0,5 días	vie 20/01/23	lun 23/01/23		
Informe de pruebas de la iteración anterior	0,5 días	vie 20/01/23	lun 23/01/23		Maria
4 Requisitos	2 días	lun 23/01/23	mié 25/01/23		
Validación de permanencia de los requisitos	2 días	lun 23/01/23	mié 25/01/23	111	Maria
4 Análisis	2 días	mié 25/01/23	vie 27/01/23		
Unificar la relación de los CU en todos los subsistemas	2 días	mié 25/01/23	vie 27/01/23	113	
4 Diseño	2 días	lun 30/01/23	mar 31/01/23		
Unificación de las clases de diseño del subsistema 1	2 días	lun 30/01/23	mar 31/01/23	111	Maria
4 Implementación	8 días	mié 01/02/23	vie 10/02/23		
Integrar componentes del subsistema 1	3 días	mié 01/02/23	vie 03/02/23	115	Maria
Implementación de los componentes separados del subsistema 2	3 días	lun 06/02/23	mié 08/02/23	113	Maria
Implementación de la BD para el subsistema 2	2 días	jue 09/02/23	vie 10/02/23	120	Maria



4 Pruebas	6 días	lun 13/02/23	lun 20/02/23		
Prueba de integración del subsistema 1 con SQL	2 días	lun 13/02/23	mar 14/02/23	119	Maria
Pruebas unitarias para cada componente JSON del subsistema 2	2 días	mié 15/02/23	jue 16/02/23	120;121	Maria
Pruebas unitarias de cada componente del subsistema 2 en SQL	2 días	vie 17/02/23	lun 20/02/23	124;120	Maria
Hito iteración 2	0 días	lun 20/02/23	lun 20/02/23	109	
4 Iteración 3	25 días	lun 20/02/23	lun 27/03/23	126	
4 Modelado de negocio	1,13 días	lun 20/02/23	mié 22/02/23		
Informe de pruebas de la iteración anterior	1,13 días	lun 20/02/23	mié 22/02/23		Maria
4 Requisitos	2 días	mié 22/02/23	vie 24/02/23		
Validación de permanencia de los requisitos	2 días	mié 22/02/23	vie 24/02/23	129	Maria
4 Análisis	3 días	lun 27/02/23	mié 01/03/23		
Refinamiento de la unificación de la realización de los CU en todos los subsistemas	3 días	lun 27/02/23	mié 01/03/23		Maria



4 Diseño	2 días	jue 02/03/23	vie 03/03/23		
Unificación de las clases de diseño del subsistema 2	2 días	jue 02/03/23	vie 03/03/23	129	Maria
4 Implementación	7 días	lun 06/03/23	mar 14/03/23		
Integrar componentes del subsistema 2	2 días	lun 06/03/23	mar 07/03/23	135	Maria
Implementación de los componentes separados del subsistema 3	2 días	mié 08/03/23	jue 09/03/23	133	Maria
Implementación de la BD para el subsistema 3	3 días	jue 09/03/23	mar 14/03/23	138	Maria
4 Pruebas	9 días	mié 15/03/23	lun 27/03/23		
Prueba de integración del subsistema 2 con SQL	2 días	mié 15/03/23	jue 16/03/23	137	Maria
Pruebas unitarias para cada componente JSON del subsistema 3	2 días	sáb 18/03/23	mar 21/03/23	138	Maria
Pruebas unitarias de cada componente del subsistema 3 en SQL	3 días	jue 23/03/23	lun 27/03/23	139;141	Maria



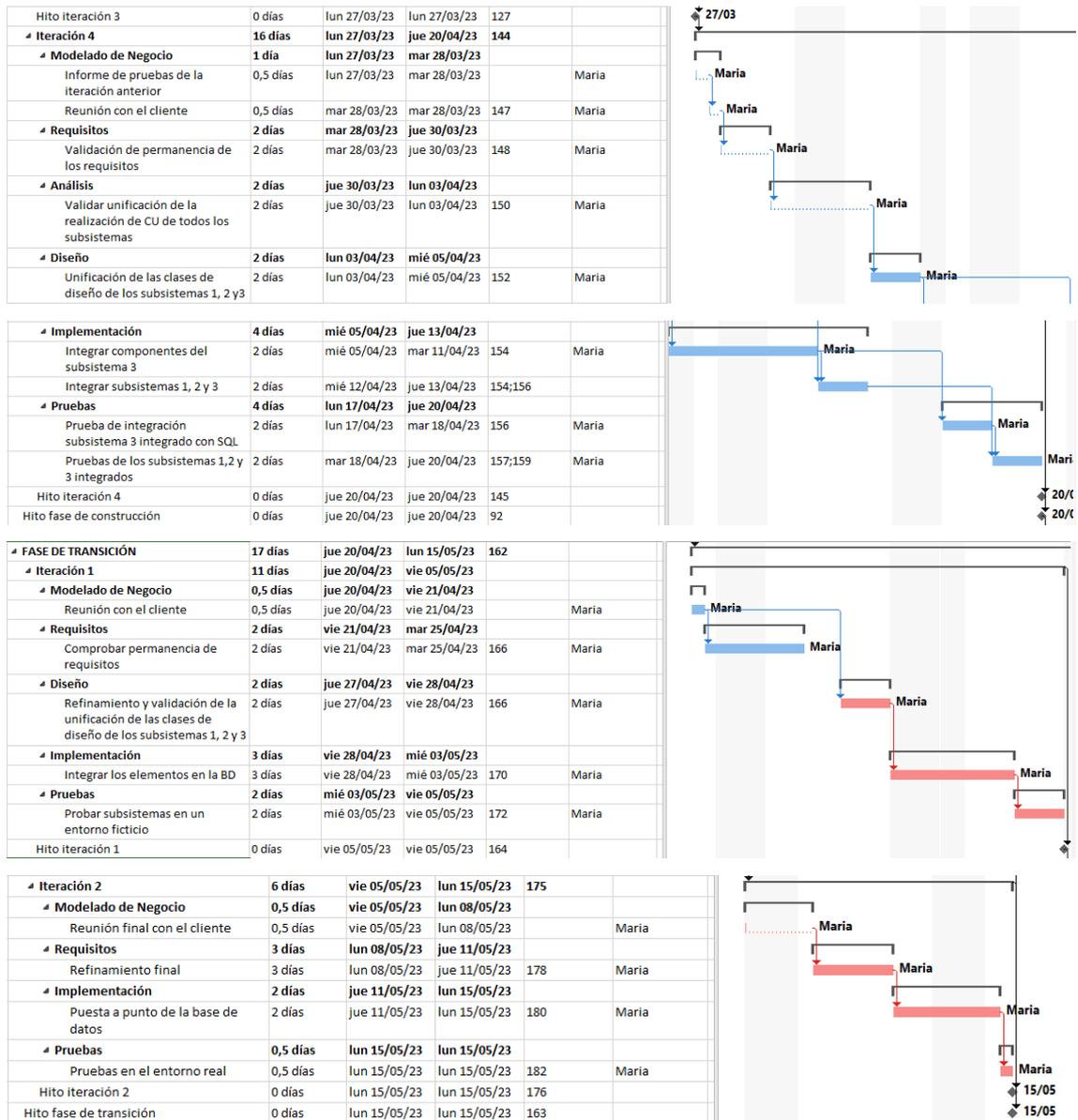


ILUSTRACIÓN 10: DIAGRAMA DE GANTT

El camino crítico se presenta de manera muy clara, resaltando las tareas críticas que están señaladas en color rojo. Cualquier retraso en estas tareas fundamentales conllevaría a una prolongación en la fecha de finalización del proyecto.

El proyecto comienza el 10 de octubre de 2022 y finaliza el 15 de mayo de 2023, por lo tanto, la duración del proyecto va a durar 7 meses, como podemos ver en las estadísticas resultantes:

Estadísticas del proyecto 'PlanificacionFinal'			
	Comienzo		Fin
Actual	lun 10/10/22		lun 15/05/23
Previsto	NOD		NOD
Real	NOD		NOD
Variación	0d		0d
	Duración	Trabajo	Costo
Actual	154d	682h	0,00 €
Previsto	0d	0h	0,00 €
Real	0d	0h	0,00 €
Restante	154d	682h	0,00 €
Porcentaje completado:			
Duración: 0%		Trabajo: 0%	
			Cerrar

ILUSTRACIÓN 11: ESTADÍSTICAS PLANIFICACIÓN TEMPORAL

Como conclusión, es evidente que los resultados muestran una similitud considerable en la cantidad de días obtenidos tanto con EZEstimate como con Microsoft Project:

Estimación	Días
EZEstimate	144,54
Microsoft Project	154

TABLA 1: COMPARATIVA DE ESTIMACIONES

Esta comparativa de estimaciones, presentada en la *Tabla 1*, revela una notable proximidad entre los valores obtenidos. Este hallazgo refuerza nuestro compromiso de alinear y cumplir con la estimación realizada en Microsoft Project. Dicha alineación se basa en los datos y análisis previamente llevados a cabo en EZEstimate. Este enfoque estratégico no solo valida nuestra metodología, sino que también sienta una base sólida para futuras implementaciones y refinamientos en el proceso de estimación.

M.6.3 Especificación de requisitos:

En esta sección, procederemos a llevar a cabo la materialización de los requisitos específicos del proyecto. Para lograr este propósito, adoptaremos la metodología desarrollada por Durán y Bernárdez. Asimismo, para la visualización y creación de los correspondientes diagramas, aprovecharemos la plataforma Visual Paradigm.

Esta metodología nos conducirá en la elaboración exhaustiva de los requisitos y metas del proyecto. Además, nos brindará una guía precisa para identificar a todos los actores involucrados en el proceso.

A continuación, delinearemos de manera concisa las fases fundamentales que abarca la especificación de requisitos. Si se requiere la información completa acerca de este proceso, se recomienda acudir al *Anexo II-Especificación de Requisitos*.

M.6.3.1 Objetivos del sistema

Como mencionamos previamente, los objetivos primordiales del sistema, crucial para la adecuada ejecución del proyecto, son los siguientes:

- Gestión de autenticación.
- Gestión de usuarios.
- Gestión de comidas.
- Gestión de ejercicios.
- Gestión de calorías.
- Gestión de progresos.
- Gestión de estadísticas.

Vamos a analizar dos presentaciones que describen los objetivos principales que el sistema debe cumplir:

OBJ-003	Gestión de comidas
Versión	1.0
Autor	María Pérez Morales.
Descripción	El sistema facilitará la visualización, adición y modificación de las diversas opciones alimenticias que se incorporan en el plan nutricional personalizado del usuario.
Subobjetivos	Ninguno
Importancia	Vital
Urgencia	Alta
Estado	Verificado
Estabilidad	Alta
Comentarios	-

TABLA 2: OBJ-GESTIÓN DE COMIDAS

OBJ-004	Gestión de ejercicios
Versión	1.0
Autor	María Pérez Morales.
Descripción	El sistema facilitará la visualización, incorporación y ajuste de los diversos ejercicios que conforman el plan de entrenamiento personalizado del usuario.
Subobjetivos	Ninguno
Importancia	Vital
Urgencia	Alta
Estado	Verificado
Estabilidad	Alta
Comentarios	-

TABLA 3: OBJ-GESTIÓN DE EJERCICIOS

M.6.3.2 Requisitos de información

Se definen requisitos para la información que el sistema almacenará, con el fin de alcanzar los objetivos establecidos en la sección anterior:

- Información general usuario.
- Información específica usuario.
- Información usuarios perfil.
- Información comidas.
- Información ejercicios.
- Información estadísticas.
- Información progresos.

A continuación, se describirán dos requisitos fundamentales de información necesarios para garantizar el correcto funcionamiento del sistema:

IRQ-004	Información comidas
Versión	1.0
Autor	María Pérez Morales.
Objetivos asociados	Tabla 6: OBJ-003.- Gestión de comidas
Descripción	El sistema está diseñado para almacenar información sobre los alimentos, específicamente en función de la dieta seleccionada por el usuario o las entradas agregadas de manera manual. Específicamente:
Datos específicos	<ul style="list-style-type: none"> • Nombre comida • Kcal
Importancia	Vital
Urgencia	Alta
Estado	Verificado
Estabilidad	Alta
Comentarios	-

TABLA 4: IRQ-INFORMACIÓN COMIDAS

IRQ-005	Información ejercicios
Versión	1.0
Autor	María Pérez Morales.
Objetivos asociados	Tabla 7 : OBJ-004.- Gestión de ejercicios
Descripción	El sistema se encargará de conservar los datos correspondientes tanto a los ejercicios predefinidos como a aquellos que se añadan manualmente. Para ser más precisos:
Datos específicos	<ul style="list-style-type: none"> • Nombre ejercicio • Kcal por minuto
Importancia	Vital
Urgencia	Alta
Estado	Verificado
Estabilidad	Alta
Comentarios	-

TABLA 5: IRQ-INFORMACIÓN EJERCICIOS

M.6.3.3 Requisitos no funcionales

Como se mencionó previamente, los requisitos no funcionales son aquellos aspectos que no están directamente ligados a las funcionalidades del sistema, sino más bien a su desempeño y características subyacentes. En el contexto de este sistema en particular, se han definido una serie de requisitos no funcionales que son fundamentales para garantizar su adecuado rendimiento y utilidad:

- Multiplataforma.
- Usabilidad.
- Concurrencia.
- Funcionamiento en tiempo real.
- Seguridad de datos.

Dentro del conjunto de requisitos no funcionales, uno se destaca como esencial para garantizar el funcionamiento óptimo del sistema: la Usabilidad. Esta cualidad se relaciona con la capacidad del sistema para ser comprensible y de fácil uso para los usuarios, sin importar su nivel de experiencia técnica. Su carácter esencial radica en el público diverso al que se dirige la aplicación, requiriendo que sea accesible y comprensible para todos los usuarios. A continuación, se va a representar detalladamente:

NFR-002	Usabilidad
Versión	1.0
Autor	María Pérez Morales.
Dependencias	Ninguna
Descripción	El sistema debe ofrecer una experiencia intuitiva y sencilla para el usuario, garantizando que los usuarios puedan navegar de manera eficaz, eficiente y sin complicaciones, permitiendo una experiencia de usuario amigable y satisfactoria.
Importancia	Alta
Urgencia	Media
Estado	Verificado
Estabilidad	Alta
Comentarios	-

TABLA 6: NFR-USABILIDAD

M.6.3.4 Requisitos funcionales

Los requisitos funcionales son elementos esenciales que detallan de manera específica las operaciones y actividades que un sistema o sus componentes deben llevar a cabo. Para lograr una comprensión más clara, procederemos a definir los conceptos de paquetes, actores y casos de uso utilizando diagramas con la herramienta Visual Paradigm.

M.6.3.4.1 Diagrama de paquetes

El diagrama de paquetes es una representación visual que ilustra los distintos elementos que constituyen el modelo global de un sistema. Este diagrama es especialmente útil para comprender la organización interna del sistema y las conexiones entre sus componentes. Su representación visual:

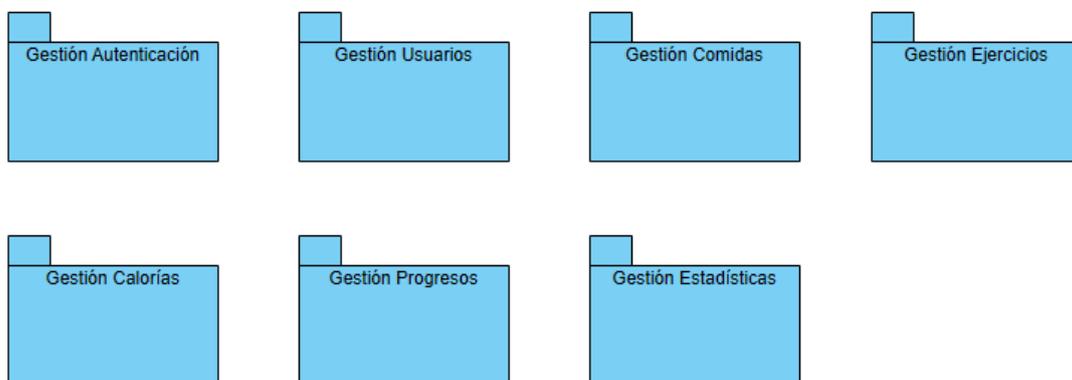


ILUSTRACIÓN 12: DIAGRAMA DE PAQUETES

M.6.3.4.2 Diagrama de actores

En esta sección se procede a definir los diferentes actores que participan en el sistema, tal como se presenta en la siguiente ilustración:



ILUSTRACIÓN 13: DIAGRAMA DE ACTORES

- **Usuario sin loguear:** este actor representa a un usuario que aún no ha realizado el proceso de autenticación. Aunque este usuario demuestra interés en acceder a las diversas funcionalidades del sistema, sus acciones se encuentran restringidas al uso de las características de acceso público que no requieren un inicio de sesión. El acceso completo a todas las capacidades del sistema se reserva hasta que el usuario complete exitosamente el proceso de registro y autenticación.
- **Usuario logueado:** este actor representa un usuario que ha completado satisfactoriamente el proceso de autenticación y ha iniciado sesión en el sistema. Esto implica acceder y utilizar todas las funcionalidades disponibles en el sistema.

M.6.3.4.3 Casos de uso

Un caso de uso se define como la presentación detallada de una acción o actividad ejecutada por ciertos actores dentro de un sistema. Procederemos a ilustrar un ejemplo de las tareas y acciones realizadas por los actores en el paquete de Gestión de comidas:

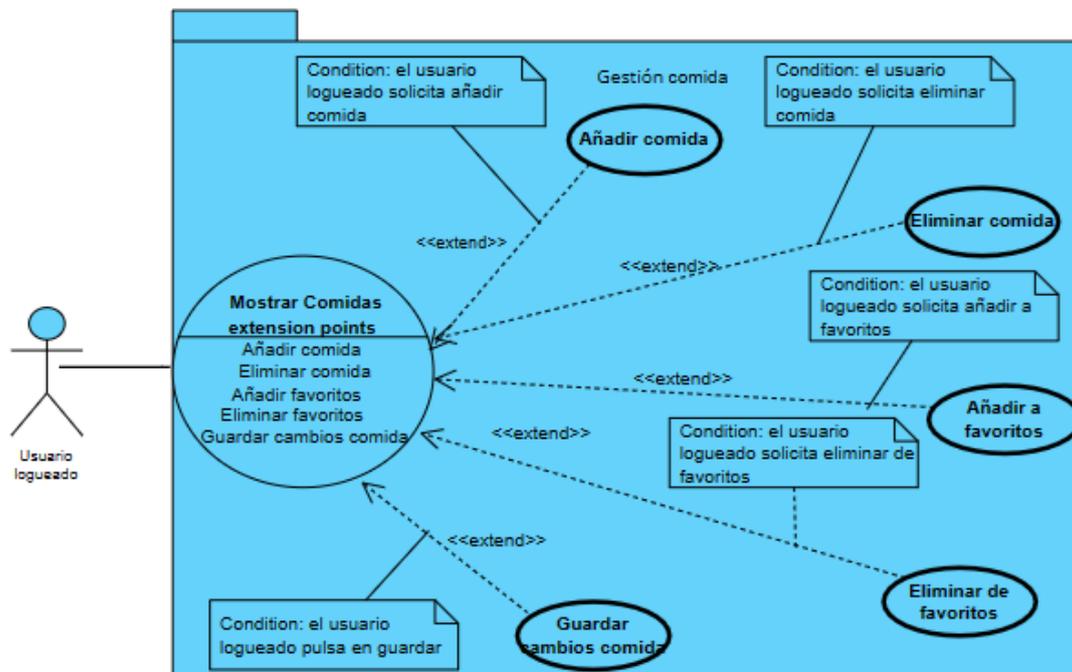


ILUSTRACIÓN 14: DIAGRAMA CASOS DE USO DEL PAQUETE GESTIÓN DE COMIDAS

UC-007	Mostrar comidas	
Versión	1.0	
Autor	María Pérez Morales.	
Dependencias	OBJ-003.- Gestión de comidas IRQ-004.- Información comidas	
Descripción	El sistema debe cumplir con las especificaciones detalladas en el caso de uso siguiente cuando el actor ACT-001.- Usuario logueado solicite al sistema ver los distintos alimentos.	
Precondiciones	El ACT-001.- Usuario logueado debe haberse registrado anteriormente.	
Secuencia	Paso	Acción
	1	El actor ACT-001.- Usuario logueado solicitará al sistema ver el listado de comidas.
	2	El sistema le mostrará el listado de alimentos que tiene guardados.
Postcondiciones		
Excepciones	Paso	Acción
	-	-
Frecuencia esperada	Alta	
Importancia	Vital	
Urgencia	Alta	
Estado	Verificado	
Estabilidad	Alta	
Comentarios	-	

TABLA 7:UC-MOSTRAR COMIDAS

UC-010	Añadir Comida	
Versión	1.0	
Autor	María Pérez Morales.	
Dependencias	OBJ-003.- Gestión de comidas IRQ-004.- Información comidas	
Descripción	El sistema deberá comportarse tal como se describe en el caso de uso siguiente cuando el actor ACT-001.- Usuario logueado solicite añadir nueva comida.	
Precondiciones	El usuario logueado debe haber realizado el caso de uso UC-009:- Mostrar Comidas	
Secuencia	Paso	Acción
	1	El actor ACT-001.- Usuario logueado solicita añadir nueva cantidad en una comida.
	2	El sistema le dará una opción para añadir campo.
	3	El actor ACT-001.- Usuario logueado añadirá nueva cantidad a un campo de comida.
	4	El sistema almacena la nueva cantidad añadida de ese alimento.
Postcondiciones	La nueva cantidad en un alimento ha sido modificada.	
Excepciones	Paso	Acción
	-	-
Frecuencia esperada	Alta	
Importancia	Vital	
Urgencia	Alta	
Estado	Verificado	
Estabilidad	Alta	
Comentarios	-	

TABLA 8: UC-AÑADIR COMIDA

UC-009	Eliminar comida											
Versión	1.0											
Autor	María Pérez Morales.											
Dependencias	OBJ-003.- Gestión de comidas IRQ-004.- Información comidas											
Descripción	El sistema debe cumplir con las especificaciones detalladas en el caso de uso siguiente cuando el actor ACT-001.- Usuario logueado solicite eliminar comida.											
Precondiciones	El ACT-001.- Usuario logueado debe haber realizado el caso de uso UC-007:- Mostrar Comidas y UC-008:- Añadir Comida.											
Secuencia	<table border="1"> <thead> <tr> <th>Paso</th> <th>Acción</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>El actor ACT-001.- Usuario logueado realizará una petición al sistema para eliminar cantidad en una comida.</td> </tr> <tr> <td>2</td> <td>El sistema le mostrará una opción para eliminar campo.</td> </tr> <tr> <td>3</td> <td>El actor ACT-001.- Usuario logueado eliminará una cantidad a un campo de comida.</td> </tr> <tr> <td>4</td> <td>El sistema almacenará la nueva cantidad añadida de ese alimento.</td> </tr> </tbody> </table>	Paso	Acción	1	El actor ACT-001.- Usuario logueado realizará una petición al sistema para eliminar cantidad en una comida.	2	El sistema le mostrará una opción para eliminar campo.	3	El actor ACT-001.- Usuario logueado eliminará una cantidad a un campo de comida.	4	El sistema almacenará la nueva cantidad añadida de ese alimento.	
Paso	Acción											
1	El actor ACT-001.- Usuario logueado realizará una petición al sistema para eliminar cantidad en una comida.											
2	El sistema le mostrará una opción para eliminar campo.											
3	El actor ACT-001.- Usuario logueado eliminará una cantidad a un campo de comida.											
4	El sistema almacenará la nueva cantidad añadida de ese alimento.											
Postcondiciones	La cantidad en un alimento ha sido modificada.											
Excepciones	<table border="1"> <thead> <tr> <th>Paso</th> <th>Acción</th> </tr> </thead> <tbody> <tr> <td>-</td> <td>-</td> </tr> </tbody> </table>	Paso	Acción	-	-							
Paso	Acción											
-	-											
Frecuencia esperada	Alta											
Importancia	Vital											
Urgencia	Alta											
Estado	Verificado											
Estabilidad	Alta											
Comentarios	-											

TABLA 9: UC-ELIMINAR COMIDA

UC-010	Añadir comida a favoritos	
Versión	1.0	
Autor	María Pérez Morales.	
Dependencias	OBJ-003.- Gestión de comidas IRQ-004.- Información comidas	
Descripción	El sistema debe cumplir con las especificaciones detalladas en el caso de uso siguiente cuando el actor ACT-001.- Usuario logueado solicite añadir comida a favoritos.	
Precondiciones	El ACT-001.- Usuario logueado debe haber realizado el caso de uso UC-007:- Mostrar Comidas.	
Secuencia	Paso	Acción
	1	El actor ACT-001.- Usuario logueado solicitará al sistema añadir nuevo campo de comida a favoritos.
	2	El sistema le mostrará una opción para añadir nuevo campo.
	3	El actor ACT-001.- Usuario logueado añadirá nuevo favorito.
	4	El sistema comprobará que es correcto.
Postcondiciones	En espera de comprobación del sistema.	
Excepciones	Paso	Acción
	-	-
Frecuencia esperada	Alta	
Importancia	Vital	
Urgencia	Alta	
Estado	Verificado	
Estabilidad	Alta	
Comentarios	-	

TABLA 10: UC-AÑADIR COMIDA A FAVORITOS

UC-011	Eliminar comida de favoritos	
Versión	1.0	
Autor	María Pérez Morales.	
Dependencias	OBJ-003.- Gestión de comidas IRQ-004.- Información comidas	
Descripción	El sistema debe cumplir con las especificaciones detalladas en el caso de uso siguiente cuando el actor ACT-001.- Usuario logueado solicite eliminar alimento de favoritos.	
Precondiciones	El ACT-001.- Usuario logueado debe haber realizado el caso de uso UC-007:- Mostrar Comidas y UC-010:- Añadir comida a favoritos .	
Secuencia	Paso	Acción
	1	El actor ACT-001.- Usuario logueado solicita al sistema eliminar campo de comida de favoritos.
	2	El sistema le mostrará la opción de eliminar de favoritos.
	3	El actor ACT-001.- Usuario logueado eliminará el alimento de favorito.
	4	El sistema comprobará que es correcto.
Postcondiciones	En espera de comprobación del sistema.	
Excepciones	Paso	Acción
	-	-
Frecuencia esperada	Alta	
Importancia	Vital	
Urgencia	Alta	
Estado	Verificado	
Estabilidad	Alta	
Comentarios	-	

TABLA 11: ELIMINAR COMIDA DE FAVORITOS

UC-011	Eliminar comida de favoritos	
Versión	1.0	
Autor	María Pérez Morales.	
Dependencias	OBJ-003.- Gestión de comidas IRQ-004.- Información comidas	
Descripción	El sistema debe cumplir con las especificaciones detalladas en el caso de uso siguiente cuando el actor ACT-001.- Usuario logueado solicite eliminar alimento de favoritos.	
Precondiciones	El ACT-001.- Usuario logueado debe haber realizado el caso de uso UC-007:- Mostrar Comidas y UC-010:- Añadir comida a favoritos .	
Secuencia	Paso	Acción
	1	El actor ACT-001.- Usuario logueado solicita al sistema eliminar campo de comida de favoritos.
	2	El sistema le mostrará la opción de eliminar de favoritos.
	3	El actor ACT-001.- Usuario logueado eliminará el alimento de favorito.
	4	El sistema comprobará que es correcto.
Postcondiciones	En espera de comprobación del sistema.	
Excepciones	Paso	Acción
	-	-
Frecuencia esperada	Alta	
Importancia	Vital	
Urgencia	Alta	
Estado	Verificado	
Estabilidad	Alta	
Comentarios	-	

TABLA 12: UC- ELIMINAR COMIDA DE FAVORITOS

M.6.4 Análisis de requisitos

El análisis de requisitos es un proceso esencial en el desarrollo de sistemas que implica examinar, refinar y organizar los requisitos del sistema previamente documentados en el *Anexo II*. Durante esta fase, se busca comprender y documentar de manera exhaustiva lo que se espera que el software haga y cumpla, para luego utilizar esta información como base para el diseño, implementación y pruebas del sistema.

Para obtener una comprensión más detallada de esta fase, se puede hacer referencia al *Anexo III*, donde se encuentra información más exhaustiva sobre el proceso de análisis de requisitos. Además, se presenta la arquitectura del modelo de análisis, lo que proporciona una comprensión visual de cómo se enlazan los diferentes elementos.

M.6.4.1 Modelo de dominio

El Modelo de Dominio se refiere a la representación visual de las clases conceptuales que componen un sistema. Estas clases se plasman a través de un diagrama de clases, el cual engloba tanto sus atributos como operaciones, además de ilustrar las interacciones que acontecen entre estos elementos. En el siguiente apartado, se presenta el Modelo de Dominio correspondiente al proyecto:

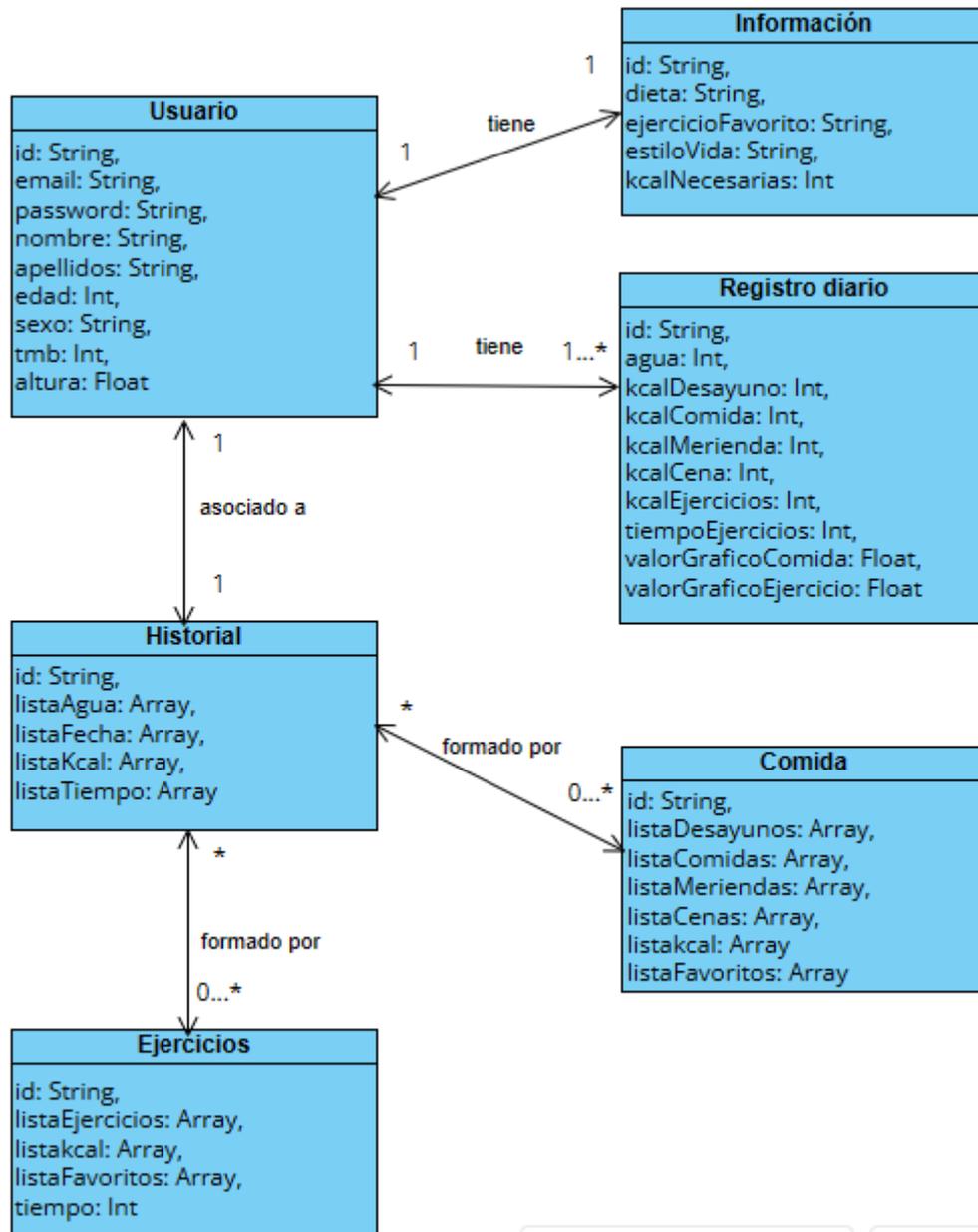


ILUSTRACIÓN 15: MODELO DE DOMINIO

M.6.4.2 Paquete de análisis

Los paquetes de análisis desempeñan un papel fundamental al ejecutar las funciones necesarias para lograr los objetivos establecidos por el sistema. Estos paquetes subdividen el sistema en módulos más manejables, lo que facilita el entendimiento y la gestión de sus componentes individuales.

La representación del sistema a través de la descomposición en paquetes de análisis consiste en visualizar y estructurar las diversas partes del sistema en bloques coherentes. Cada uno de estos paquetes de análisis agrupa elementos relacionados en función de su funcionalidad y responsabilidad. Este enfoque permite una comprensión más clara de cómo se interrelacionan las distintas partes del sistema y cómo contribuyen conjuntamente a la consecución de los objetivos generales.

El sistema se representa mediante la descomposición en paquetes de análisis:

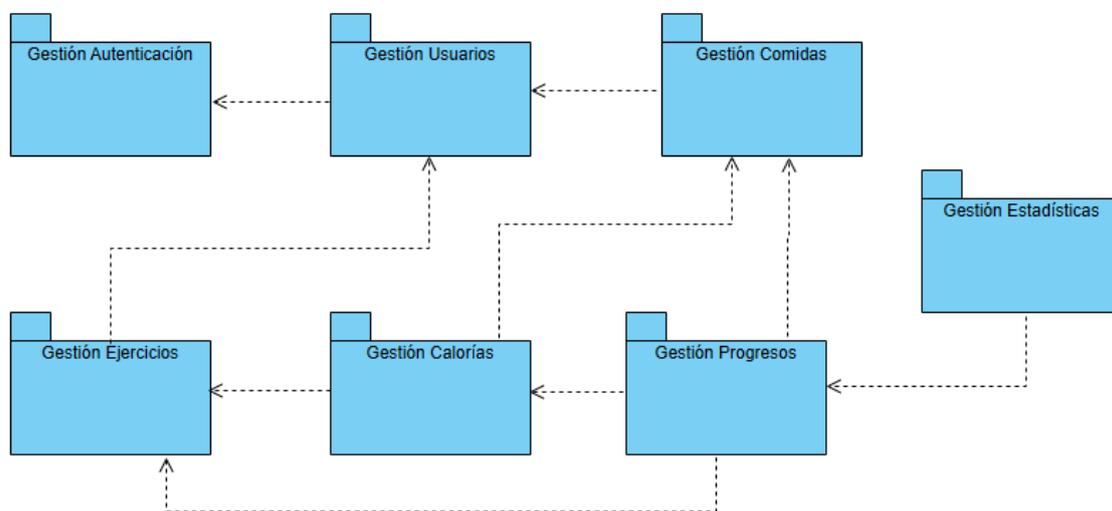


ILUSTRACIÓN 16: DIAGRAMA PAQUETE DE ANÁLISIS

M.6.4.2.1 Clases de análisis

Este punto aborda la presentación de diversas categorías y subsistemas de análisis, que se dividen en tres tipos principales: entidad, control e interfaz. Estas categorías se encuentran organizadas en paquetes, como se ilustra en el *Diagrama de Paquete de Análisis* (consultar *Ilustración 16*). Para una comprensión más clara y detallada, se puede recurrir al *Anexo III - Análisis de Requisitos*, donde se proporcionan todos los diagramas necesarios del sistema.

A continuación, detallaremos dos paquetes de análisis que el sistema a modo de ejemplo.

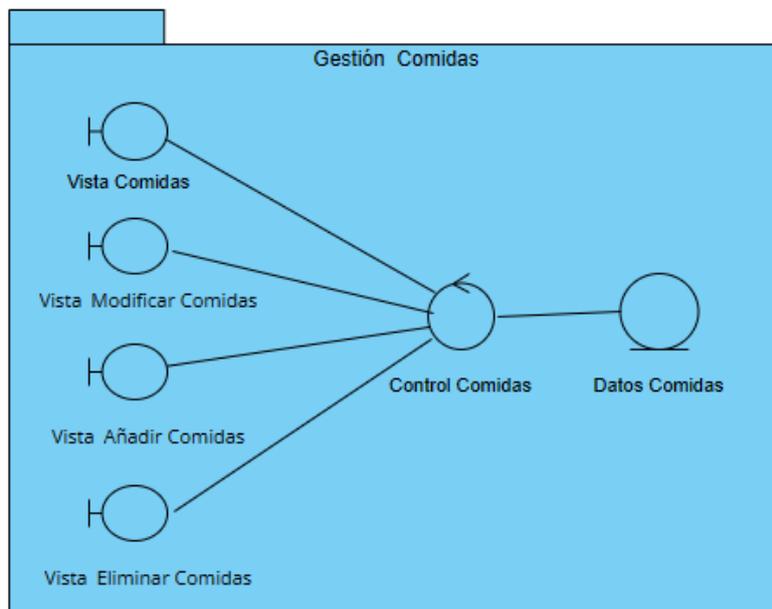


ILUSTRACIÓN 17: DIAGRAMA CLASES DE ANÁLISIS GESTIÓN COMIDAS

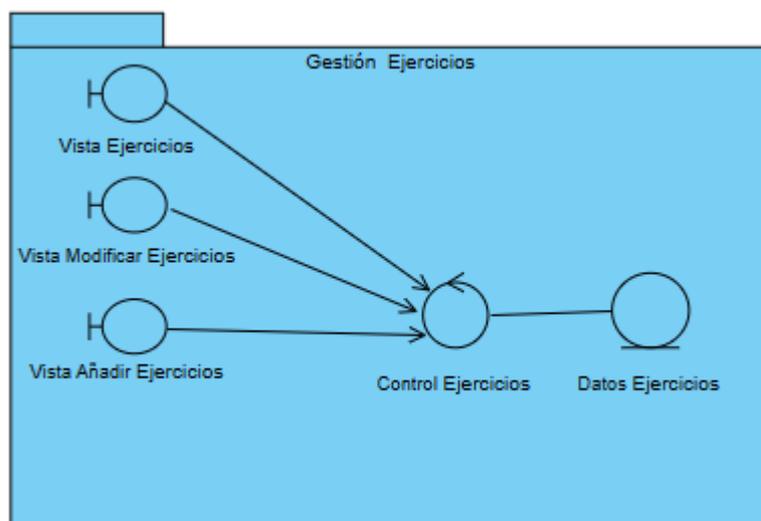


ILUSTRACIÓN 18: DIAGRAMA CLASES DE ANÁLISIS GESTIÓN EJERCICIOS

M.6.4.3 Realización de casos de uso

En este punto, presentaremos de manera gráfica las distintas interacciones que tienen lugar entre los elementos participantes dentro de los casos de uso previamente descritos. Para lograr una comprensión más detallada, emplearemos diagramas de secuencia, herramientas visuales que permiten una representación clara y precisa de la secuencia de acciones y mensajes entre estos elementos. A continuación, exploraremos dos ejemplos concretos para una mejor ilustración:

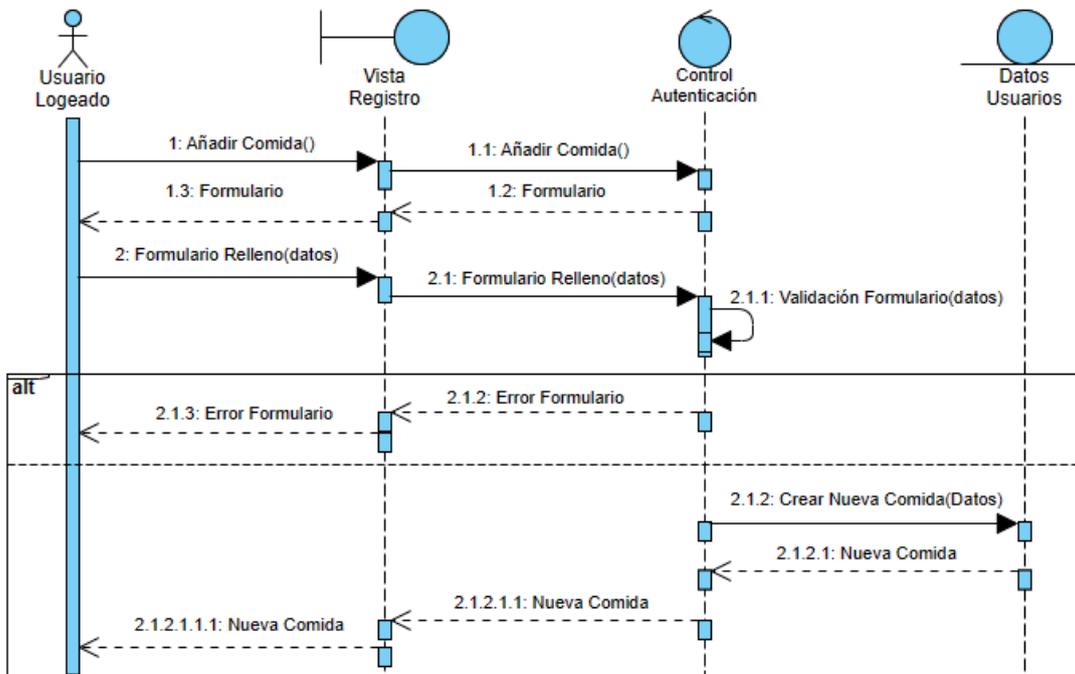


ILUSTRACIÓN 19: DIAGRAMA DE SECUENCIA UC-AÑADIR COMIDA

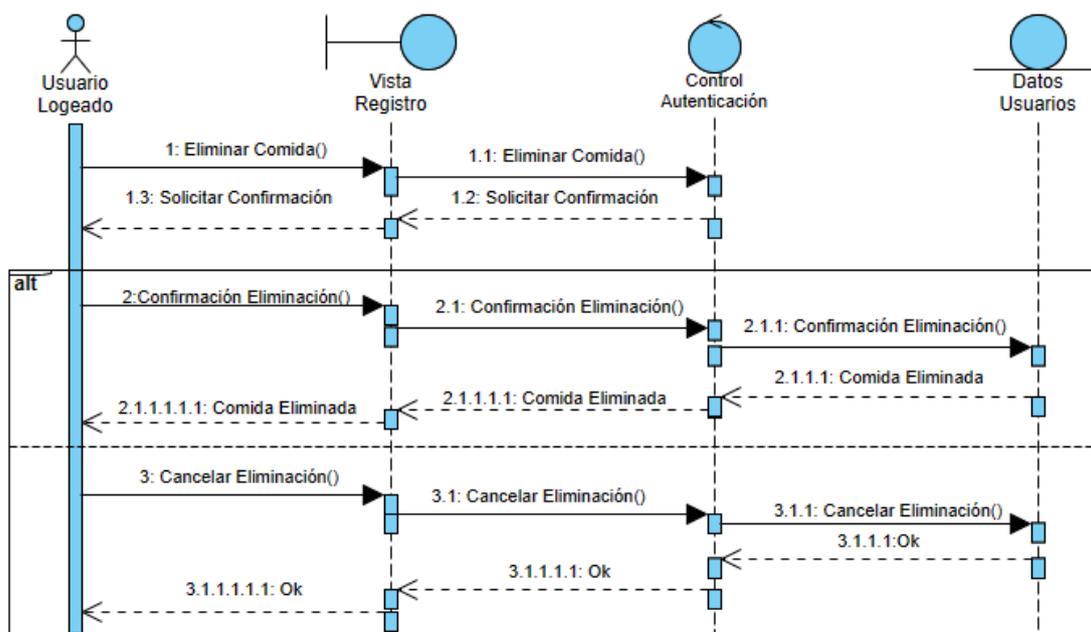


ILUSTRACIÓN 20: DIAGRAMA DE SECUANCIA UC-ELIMINAR COMIDA

M.6.5 Diseño del sistema

El diseño del sistema ofrece una exhaustiva documentación acerca de la concepción integral del sistema, lo cual simplifica el proceso de evaluación y elección de la solución más idónea.

Para consultar la información más detallada sobre el diseño del sistema consultar el *Anexo IV*.

M.6.5.1 Patrones arquitectónicos

Los Patrones Arquitectónicos se refieren a las estructuras reutilizables adoptadas durante la etapa de desarrollo de un sistema. Se emplea el patrón Modelo-Vista-VistaModelo (MVVM), el cual simplifica la realización de pruebas y el proceso de mantenimiento. MVVM contribuye a lograr una clara separación entre la lógica de negocio y la presentación en la interfaz de usuario. Esto asegura que la interfaz de usuario sea independiente de la plataforma subyacente, lo que a su vez agiliza la detección y corrección de errores. Además, este patrón facilita la adaptación y el crecimiento continuo del proyecto.

A continuación, se muestra el patrón arquitectónico:

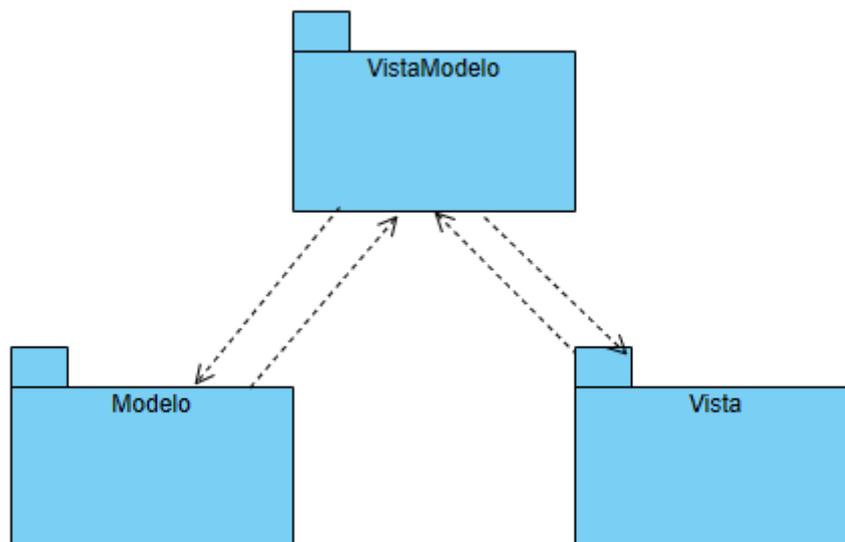


ILUSTRACIÓN 21: DIAGRAMA PATRÓN MVVM

M.6.5.2 Subsistemas de diseño

Se presentarán a continuación los distintos subsistemas de diseño, cada uno con sus respectivas definiciones, y se analizarán sus interrelaciones con el objetivo de alcanzar un propósito compartido. Esta información se encuentra ilustrada en la captura que se muestra a continuación:

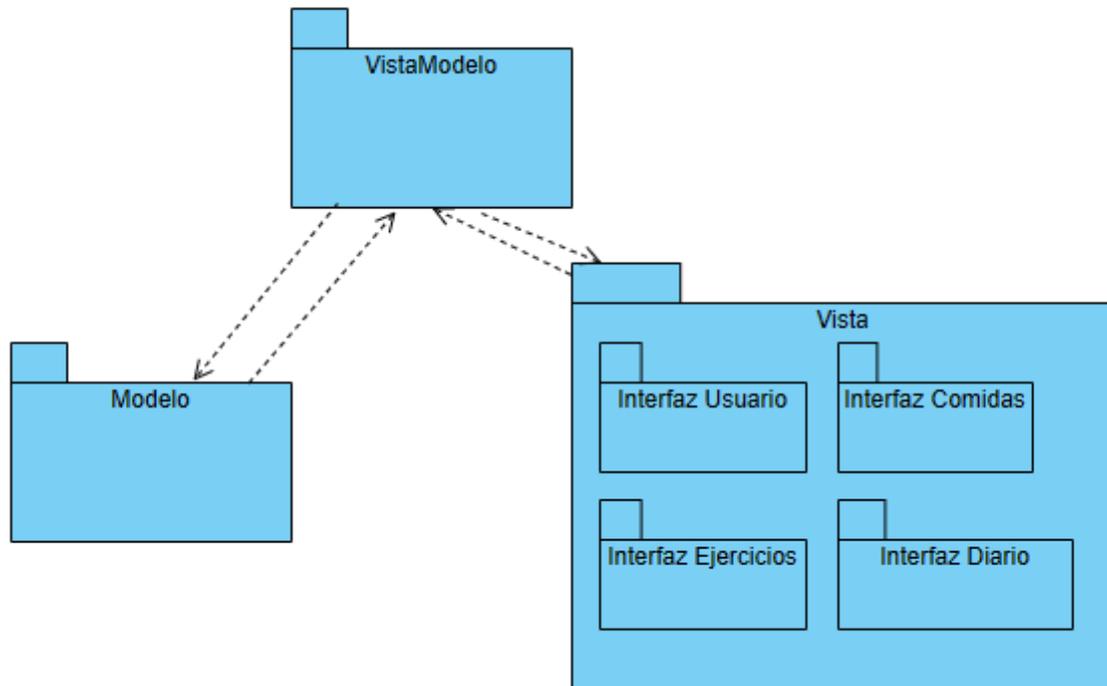


ILUSTRACIÓN 22: DIAGRAMA SUBSISTEMAS DE DISEÑO

M.6.5.3 Clases de diseño

Se presentará la capa VistaModelo, capa intermedia que actúa como un puente entre la interfaz de usuario (Vista) y los datos y lógica subyacentes (Modelo). Las capas adicionales se describen detalladamente en el *Anexo IV - Diseño del sistema software*.

Un esquema visual de esta estructura se representa en la ilustración que sigue:

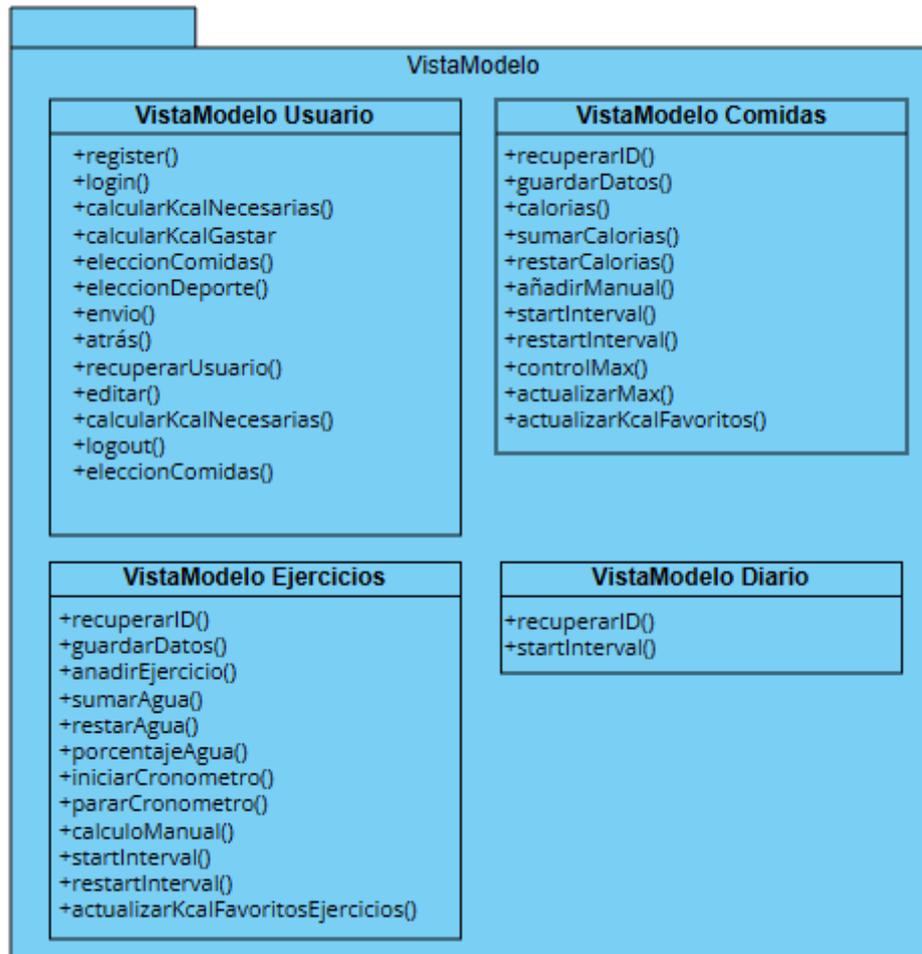


ILUSTRACIÓN 23: DIAGRAMA CLASES DE DISEÑO VISTAMODELO

M.6.5.4 Realización de casos de uso

Para llevar a cabo la realización de casos de uso se han empleado diagramas de secuencia con el propósito de visualizar de manera efectiva el flujo de interacciones entre los objetos y actores durante la realización de un caso de uso particular. Estos diagramas permiten una comprensión más clara de la dinámica del sistema en situaciones concretas, ya que muestran la secuencia de mensajes que se intercambian entre los objetos a lo largo del tiempo.

Se parte de los realizados en el *Anexo III* y se refinarán para conseguir la mayor aproximación a los métodos implementados en el proyecto.

Un ejemplo de lo anterior:

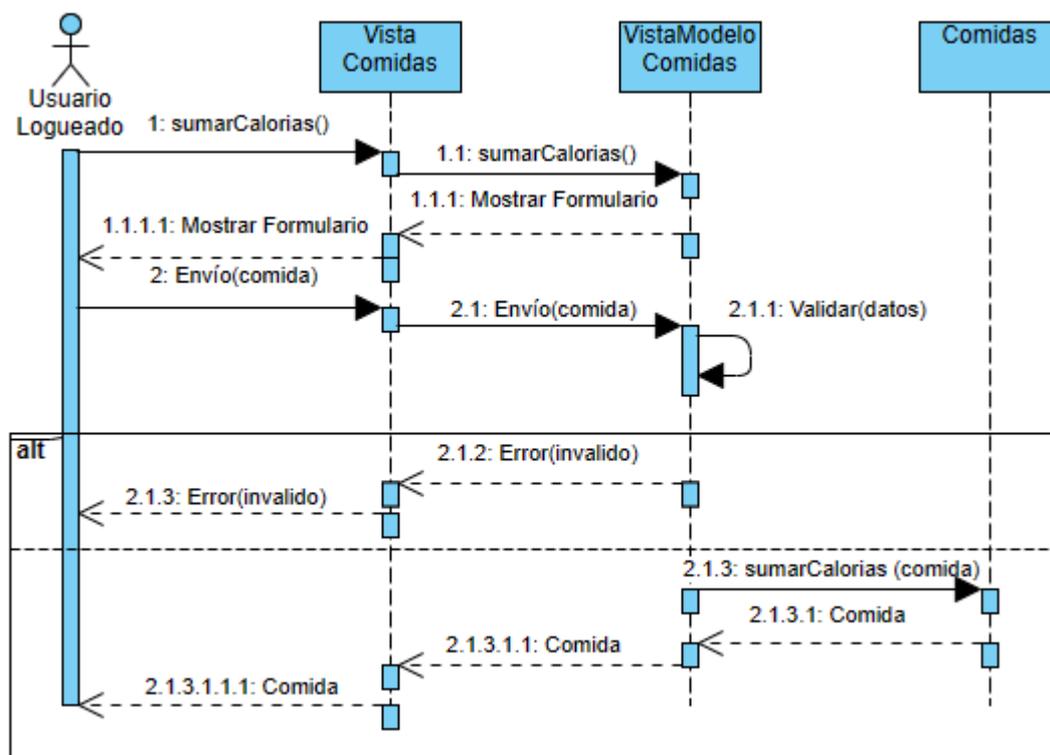


ILUSTRACIÓN 24: REALIZACIÓN CASOS DE USO UC-AÑADIR COMIDA

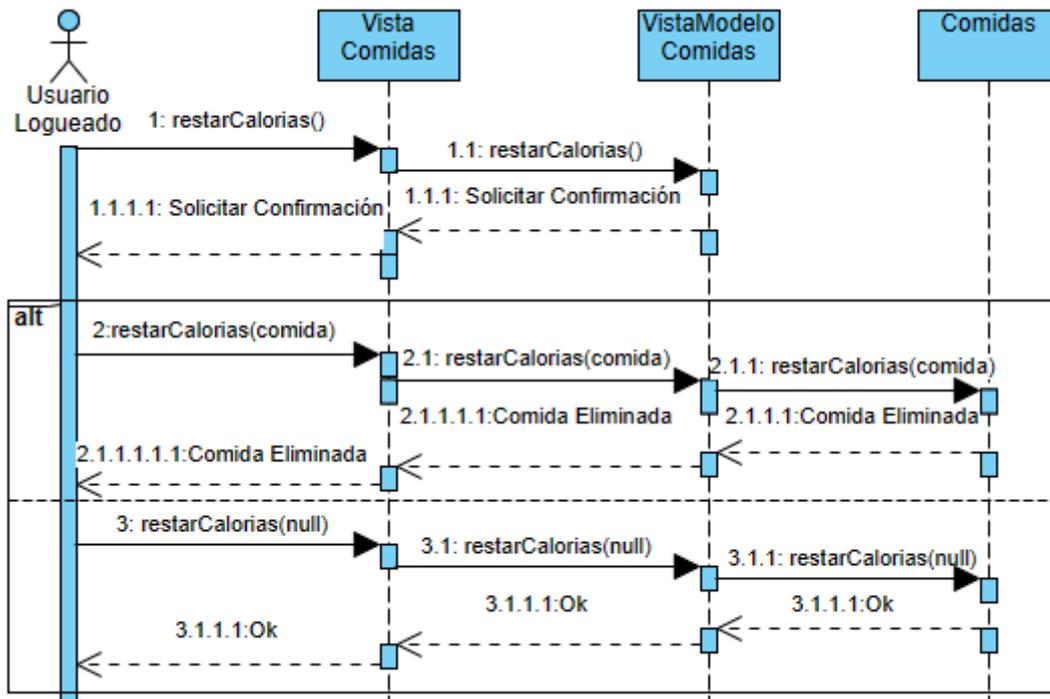


ILUSTRACIÓN 25: REALIZACIÓN CASOS DE USO UC-ELIMINAR COMIDA

M.6.5.5 Diseño de la base de datos

Para garantizar la máxima eficiencia y funcionalidad del sistema, resulta imperativo contar con una estructura de almacenamiento óptima. Este objetivo se alcanza mediante la implementación de una base de datos, tal como Firebase. Dicha base de datos desempeñará un papel crucial al alojar datos críticos, tales como información de usuarios, detalles de comidas, rutinas de ejercicio y registros diarios. Estos elementos son esenciales para asegurar el correcto funcionamiento del usuario dentro de la aplicación.

Esta base de datos estará organizada en diversas tablas que representan estos componentes fundamentales:

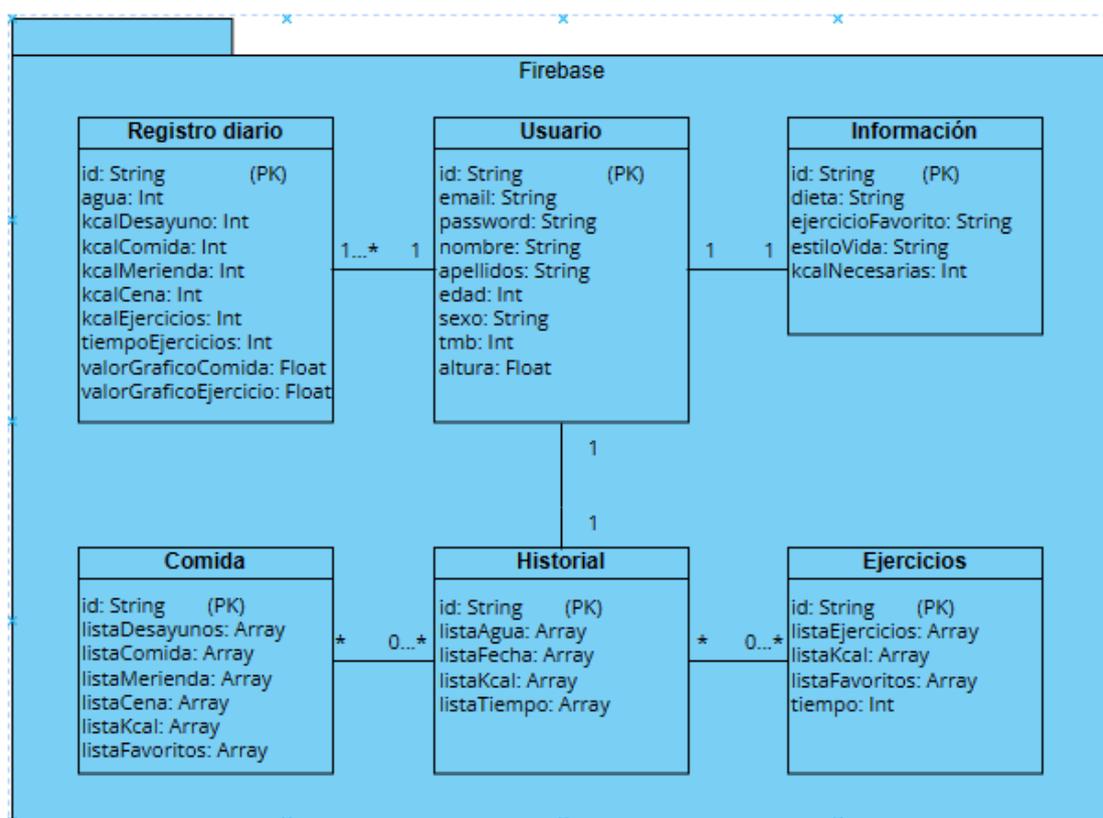


ILUSTRACIÓN 26: DISEÑO DE FIREBASE

M.6.5.6 Modelo de despliegue

El modelo de despliegue se encarga de establecer la disposición y distribución de los componentes operativos en distintos nodos, lo que a su vez describe la configuración física y topológica del sistema. La comunicación y colaboración entre estos nodos se efectúan mediante la utilización de Internet como canal de interconexión. Básicamente, este modelo proporciona una representación de cómo el sistema toma forma en un entorno concreto. En este contexto, se distinguen tres nodos fundamentales:

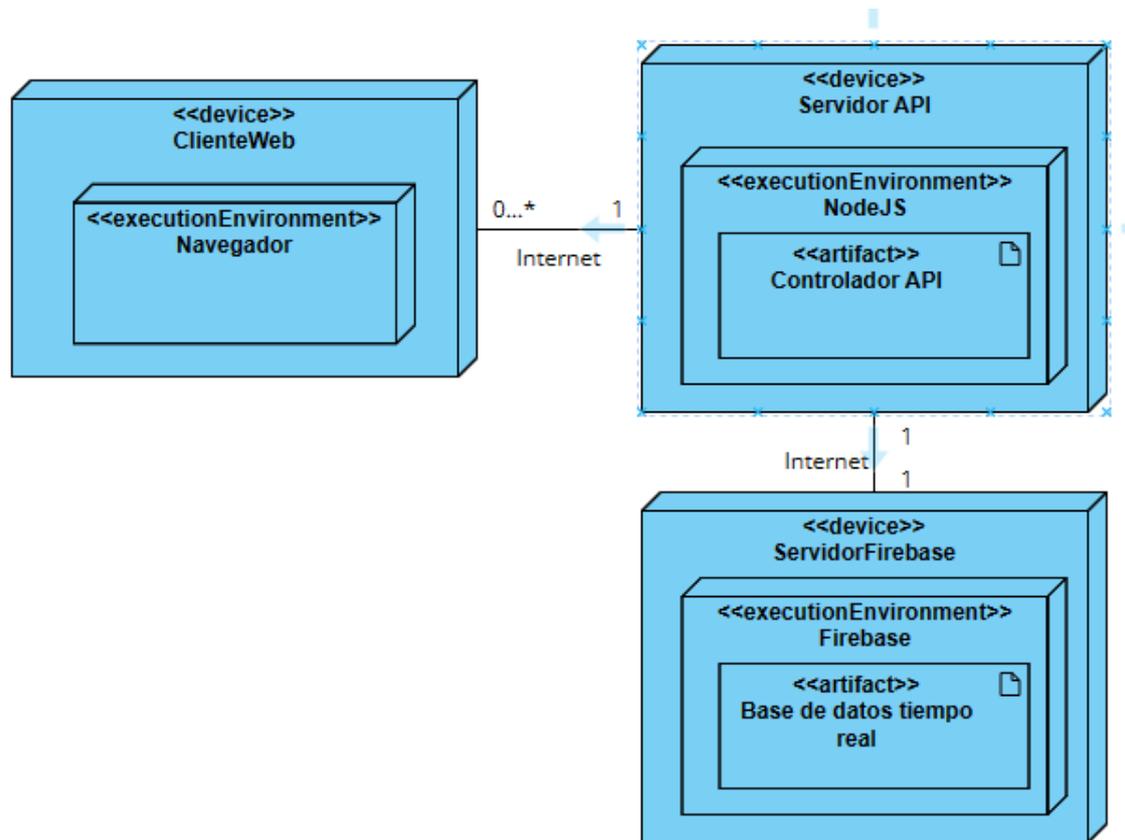


ILUSTRACIÓN 27: DIAGRAMA MODELO DE DESPLIEGUE

M.6.6 Implementación

La implementación de este proyecto implica proporcionar una guía exhaustiva del código de software desarrollado. Para facilitar la comprensión del código y garantizar que cualquier modificación futura se realice de manera eficiente.

Para obtener información más detallada sobre la implementación técnica del proyecto, se recomienda consultar el *Anexo V - Documentación Técnica*.

Durante el proceso de desarrollo, se emplearon diversas técnicas y herramientas previamente mencionadas para lograr la realización del sistema.

La estructura del sistema se compone de los siguientes elementos:

- **Interfaz:** los usuarios participan en interacciones con el sistema mediante una aplicación móvil. Esta aplicación móvil actúa como la interfaz principal a través de la cual los usuarios pueden acceder y manipular las funciones del sistema.
- **Desarrollo:** el sistema está desarrollado en Vue, un framework basado en HTML, CSS y JavaScript, en combinación con el framework Ionic.
- **Base de Datos:** la información es almacenada y recuperada mediante el uso de Firebase, que facilita la administración de datos y la interacción con los mismos. El archivo **firebase.js** es crucial en este proceso, ya que contiene la configuración esencial para establecer una conexión fluida con Firebase, incluye las credenciales necesarias para que se pueda interactuar de manera segura y autenticada con los servicios proporcionados por Firebase.

Las representaciones gráficas de la interfaz de usuario se van a ver en la carpeta **views**. Incluyen distintas pantallas / subpantallas y los métodos asociados que determinan su comportamiento y funcionalidad. Entre las pantallas principales encontramos:

- **Pantalla RegistrarUsuario.vue:** esta pantalla corresponde a un formulario diseñado para los usuarios que aún no se han registrado en la aplicación. Su propósito es facilitar el proceso de registro y acceso a la plataforma.
- **Pantalla CuentanosDeTi.vue:** similar a la anterior, esta pantalla también es un formulario dirigido a los usuarios no registrados. Su objetivo es recopilar información relevante para el registro y acceso adecuado a la aplicación.
- **Pantalla InicioSesion.vue:** destinada a los usuarios que ya se han registrado, esta pantalla permite el inicio de sesión y otorga acceso a otras funcionalidades disponibles en la aplicación.
- **Pantalla NuevaContraseña.vue:** este archivo se utiliza para el restablecimiento de contraseñas olvidadas. Proporciona un enlace de restablecimiento que se envía al correo electrónico proporcionado por el usuario.
- **Pantalla PaginaAlimentacion.vue:** diseñada con el propósito de mostrar la dieta seleccionada por el usuario, esta pantalla ofrece además otras funcionalidades relacionadas.
- **Pantalla PaginaDiario.vue:** en este archivo se gestionan las estadísticas de calorías, tanto las ingeridas a través de los alimentos como las gastadas en actividades físicas.
- **Pantalla PaginaEjercicios.vue:** se trata de un archivo dedicado a presentar una variedad de ejercicios destinados a actividades deportivas, entre otras funciones.
- **Archivo PaginaPerfil.vue:** esta representación está diseñada para permitir la visualización y edición de los datos del usuario. Realizar cambios en esta pantalla afectará la información del usuario en toda la aplicación.

M.6.7 Pruebas

Las pruebas desempeñan un papel de vital importancia en el proceso de desarrollo de sistemas de software. Su función primordial radica en verificar y garantizar el correcto funcionamiento del sistema, identificando posibles fallos que podrían haber pasado desapercibidos sin la realización de estas evaluaciones exhaustivas.

En este contexto, las pruebas cumplen un rol fundamental en la ejecución exitosa del proyecto, ya que permiten la detección temprana y la corrección de pequeños errores antes de que se conviertan en problemas más complejos. A lo largo del desarrollo del proyecto, se han llevado a cabo pruebas de manera regular y sistemática. Tras la implementación de cada componente, no solo se sometía dicho componente a pruebas individuales, sino que también se procedía a evaluar la interacción entre los distintos componentes del proyecto en su conjunto. Este enfoque amplio permitía detectar posibles implicaciones y efectos colaterales derivados de modificaciones realizadas en una parte del sistema, evitando así la propagación de errores inadvertidos.

Asimismo, al culminar la implementación integral del sistema, se realizaron múltiples pruebas exhaustivas del sistema en su estado final. Estas pruebas se llevaron a cabo con la finalidad de asegurar el correcto funcionamiento del sistema en su totalidad, antes de proceder a su entrega final.

M.6.8 Documentación

En este proyecto, se emplea JSDoc, una poderosa herramienta diseñada para la generación de documentación de código. Es importante destacar que JSDoc está especialmente orientada a la documentación de JavaScript. Su principal ventaja radica en su capacidad para generar automáticamente información técnica a partir de los comentarios incorporados en el código fuente.

Para obtener información más detallada sobre la documentación técnica, puedes consultar el *Anexo V*. Este anexo proporciona una guía exhaustiva que abarca desde la instalación de JSDoc hasta el proceso completo de generación de documentación. Además, dentro del proyecto, se ha habilitado una carpeta dedicada que alberga toda la documentación generada.

A continuación, presentamos un ejemplo concreto de documentación técnica. Enfocándonos en la vista de la página de ejercicios, se detallarán exhaustivamente las distintas funciones junto con breves descripciones para cada una de ellas:

Module: PaginaEjercicios

Default exported object containing component's data and methods.

Source: [PaginaEjercicios.vue, line 11](#)

Methods

"añadirEjercicio"()

Adds a new exercise to the user's exercise list.

Source: [PaginaEjercicios.vue, line 123](#)

actualizarKcalFavoritosEjercicios()

Updates the list of favorite exercises and their kcal.

Source: [PaginaEjercicios.vue, line 296](#)

Home

Modules

CuentanosDeTi
PaginaDiario
PaginaEjercicios
PaginaPerfil
RegistrarUsuario
firebase/auth

Global

calcularKcalNecesarias
editar
eleccionComidas
logout

ILUSTRACIÓN 28: INTERFAZ GRÁFICA DOCUMENTACIÓN TÉCNICA

M.6.9 Funcionalidad del sistema

En este punto se proporciona un desglose de las diversas pantallas con las que el usuario interactúa, junto con las subpantallas disponibles desde cada una de ellas. Además, con el objetivo de mejorar la identificación y comprensión de cada sección, se acompaña cada pantalla con su respectivo nombre o icono representativo. Esta estrategia proporciona una guía visual y descriptiva que facilitará la comprensión inmediata del funcionamiento del sistema.

Para acceder a una descripción completa y detallada de todas las funcionalidades del sistema, se recomienda consultar el *Anexo VI - Manual de Usuario*.

M.6.9.1 Autenticación

El usuario no logueado al acceder a la siguiente vista donde encontrará un formulario diseñado específicamente para iniciar sesión. Este formulario solicita generalmente información de identificación, como nombre de usuario y contraseña, para otorgar acceso a las funciones o secciones restringidas del sistema o aplicación.



ILUSTRACIÓN 29: PANTALLA INICIO DE SESIÓN

Como opción a destacar, tenemos la de recuperar la contraseña que muestra el siguiente formulario, donde el usuario solicitará una nueva contraseña a partir de su email:

Recuperar contraseña

✉ Email

ENVIAR CORREO

Atras >

ILUSTRACIÓN 30: PANTALLA RESTABLECIMIENTO CONTRASEÑA

La base de datos encargada de la autenticación del usuario, Firebase, comprobará que el email es válido y le enviará al usuario un correo como el siguiente para permitirle modificar su contraseña:

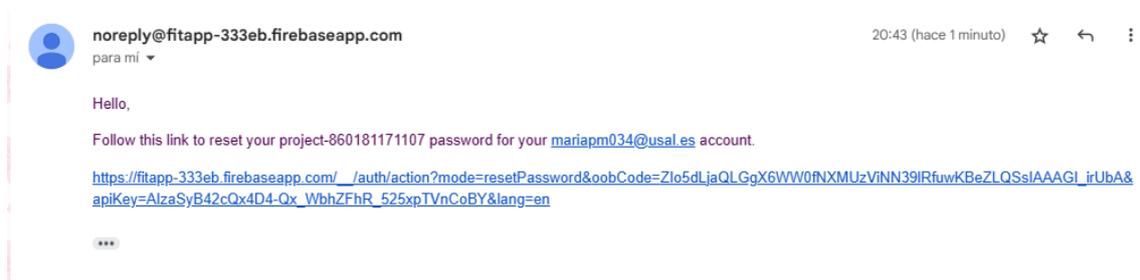


ILUSTRACIÓN 31: EMAIL MODIFICACIÓN CONTRASEÑA

M.6.9.2 Gestión de ejercicios

Después de que el usuario haya iniciado sesión en la aplicación, se le conoce como "usuario logueado". Este estado de autenticación le confiere derechos específicos para interactuar con las funcionalidades de la aplicación.

En este ejemplo el sistema permite al usuario logueado gestionar los datos de sus actividades físicas y hábitos de hidratación. Concretamente, el usuario puede:

- **Consultar el listado de ejercicios:** se muestran todos los ejercicios que el usuario ha registrado en el sistema, dependiendo de las preferencias del usuario o la configuración dentro de la aplicación.
- **Verificar la cantidad de agua consumida:** esta función refleja la cantidad de vasos de agua que el usuario ha bebido durante el día, ayudándolo a monitorizar su hidratación y garantizar un consumo adecuado de líquidos mediante un gráfico de barras.
- **Gráfico de calorías gastadas:** a través de un gráfico circular de progreso, se presenta la cantidad de calorías que el usuario ha quemado mediante sus ejercicios, lo que permite seguir su progreso y entender mejor el impacto de sus actividades físicas en su salud y bienestar.



ILUSTRACIÓN 32: PANTALLA DE EJERCICIOS

Como opción a destacar, tenemos la de añadir una actividad. La cual tenemos dos opciones.

- **Iniciar el cronómetro:** al seleccionar esta opción, se activa un cronómetro que se exhibe en la interfaz de usuario, como se ilustra a continuación. Una vez iniciado el cronómetro, se habilita un segundo botón de pausa, permitiendo al usuario detener el temporizador en cualquier momento. Este enfoque es útil para medir con precisión la duración de la actividad deportiva en tiempo real.

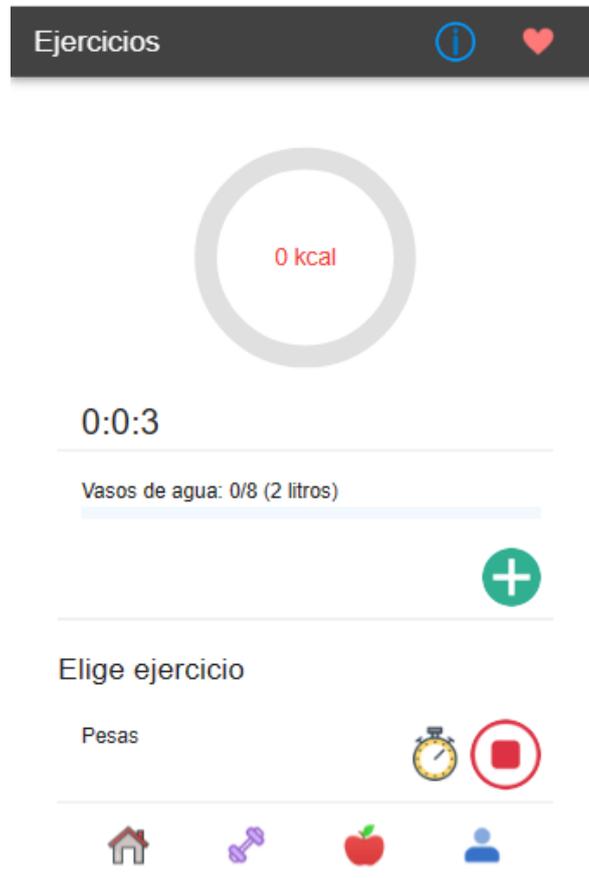


ILUSTRACIÓN 33: PANTALLA AÑADIR EJERCICIO CRONÓMETRO

- **Añadir manualmente:** al optar por esta alternativa, el usuario accede a la subpantalla mostrada. Aquí, se despliega un formulario interactivo que solicita los datos esenciales para registrar el ejercicio realizado y calcular las calorías quemadas. El formulario incluye campos para ingresar información como el tipo de actividad y la duración.

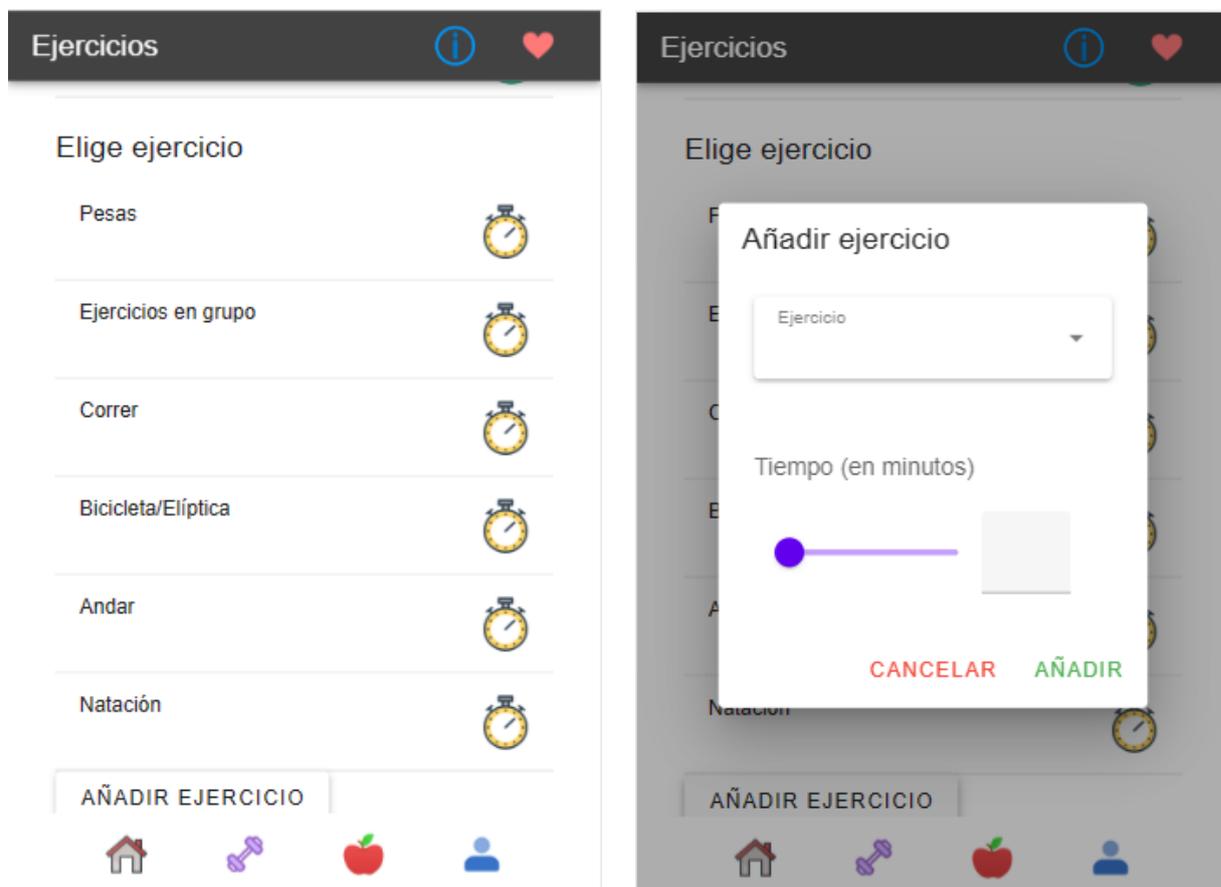


ILUSTRACIÓN 34: PANTALLA AÑADIR EJERCICIO MANUALMENTE

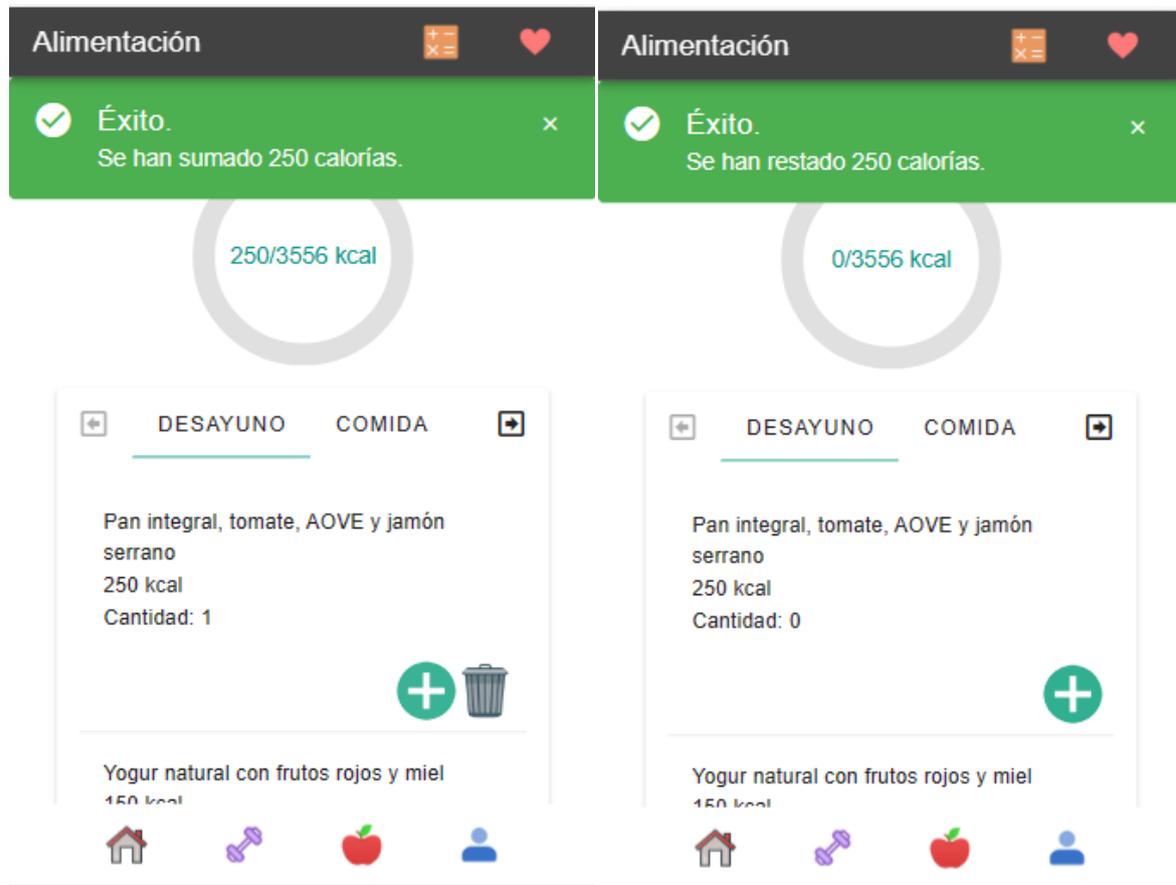
M.6.9.3 Gestión de comidas

El sistema permite al usuario logueado gestionar los datos de las comidas, brindando al usuario la posibilidad de acceder y explorar las distintas opciones de comidas almacenadas en el sistema. Con el propósito de ofrecer una experiencia organizada, se han establecido cuatro listas principales: la lista correspondiente al desayuno, la lista de comida, la de merienda y, por último, la lista dedicada a las cenas, como vemos a continuación:



ILUSTRACIÓN 35: PANTALLA DE COMIDAS

Como opción a destacar, tenemos la de añadir comida. Esta funcionalidad habilita al usuario para registrar una nueva comida en su diario alimenticio, especificando la cantidad correspondiente del alimento consumido. Además, se registran automáticamente las calorías asociadas a esa comida, incrementando así el recuento total de calorías consumidas en el día. Para brindar una gestión completa, también se proporciona la opción de eliminar un alimento previamente añadido mediante el icono de basura que aparece en la siguiente ilustración:



**ILUSTRACIÓN 36: PANTALLA DE
AÑADIR/ELIMINAR COMIDA**

Además, observamos un mensaje de notificación que puede aparecer para confirmar la adición o eliminación de la comida. Este proceso también impacta en la visualización gráfica de las calorías ingeridas, ya que el gráfico se ajusta en tiempo real para reflejar la nueva información alimenticia ingresada.

M.6.9.4 Gestión de usuario

El sistema permite al usuario logueado gestionar sus datos, permitiéndole consultar o modificar sus datos, que tendrán efectos en las diferentes pantallas.

Para llevar esto a cabo, el usuario tiene la pantalla de perfil, al cual accede a sus datos:

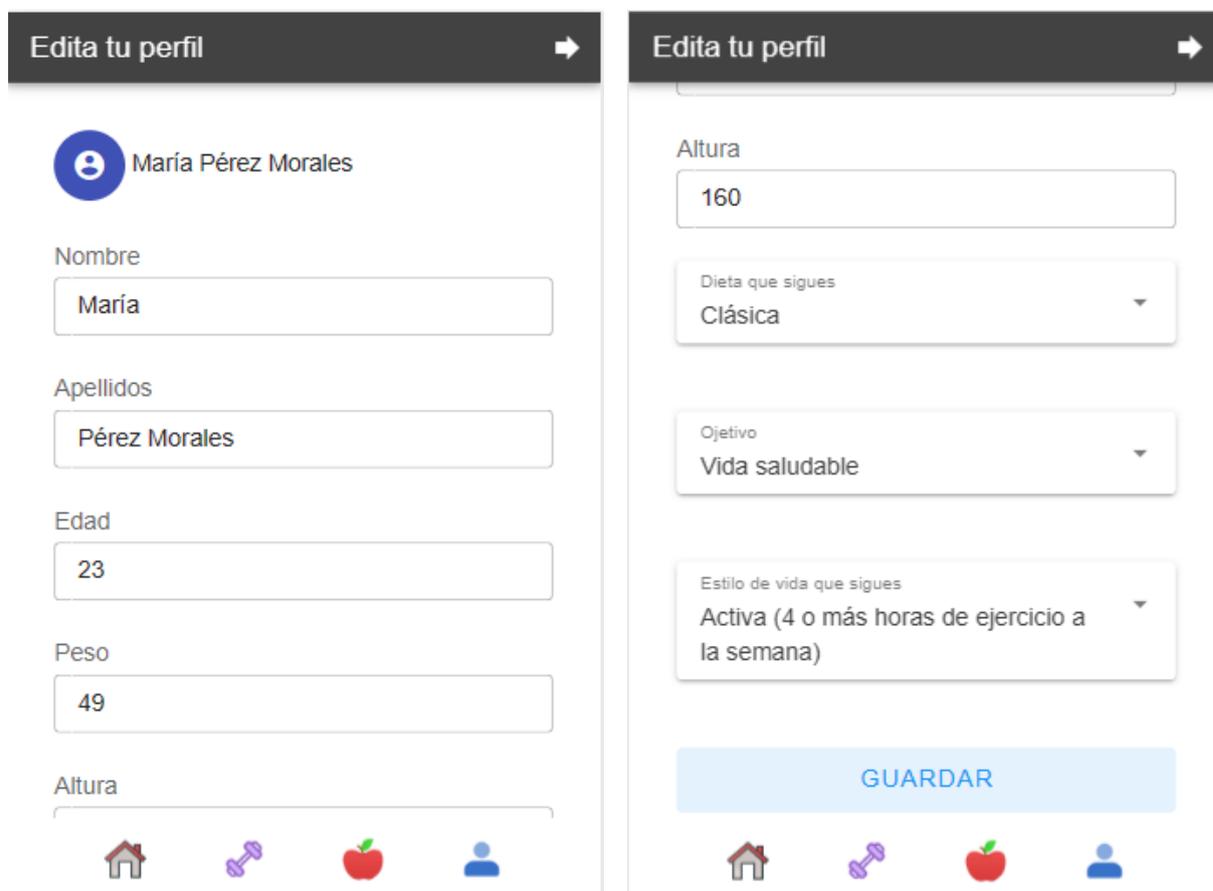


ILUSTRACIÓN 37: PANTALLA PERFIL

Una de las opciones importantes de esta pantalla, es que a partir del siguiente botón, el usuario logueado accederá a la confirmación para cerrar sesión:

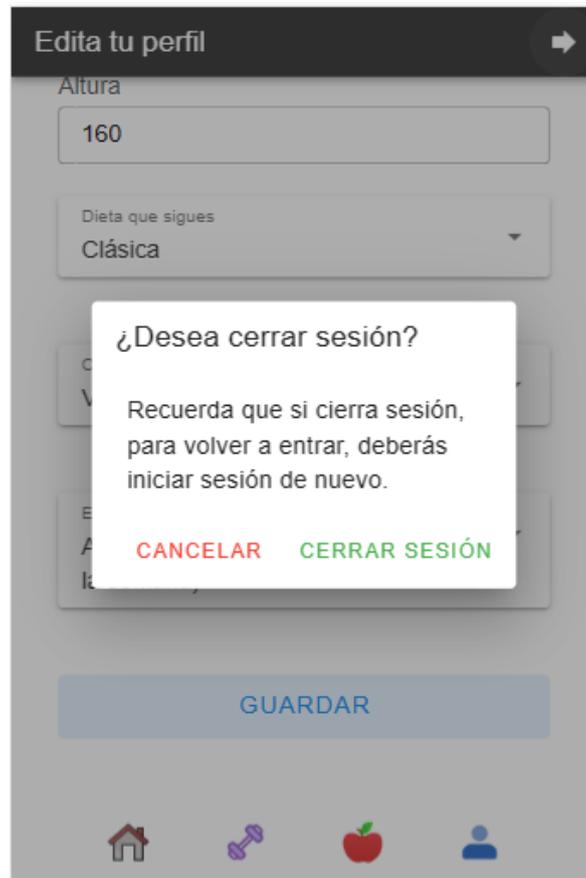


ILUSTRACIÓN 38: PANTALLA DE CERRAR SESIÓN

Se confirma acción de cerrar sesión pinchando en el botón “Cerrar sesión” que se muestra a continuación:

CERRAR SESIÓN

M.7 CONCLUSIONES

En conclusión, me siento enormemente satisfecha y orgullosa de haber culminado exitosamente este proyecto de desarrollo de una aplicación móvil integral. He logrado plenamente los objetivos que me propuse al inicio, creando una plataforma que abarca desde la autenticación y registro de usuarios hasta la gestión de registros de actividades físicas y alimentos ingeridos, todo ello con una interfaz de usuario amigable y funcional. Un logro fundamental ha sido el cumplimiento de los requisitos, tanto funcionales como no funcionales, lo que garantiza una calidad y eficiencia óptimas en el sistema.

La aplicación no solo brinda la capacidad de registrar y visualizar actividades y alimentos, sino que también capacita a los usuarios con un control total sobre sus registros, permitiéndoles realizar modificaciones y ajustes de acuerdo a sus cambiantes necesidades. Además, un aspecto destacado es que la aplicación empodera a los usuarios con la libertad de diseñar sus propias dietas personalizadas y crear ejercicios a su medida. Estas creaciones pueden ser guardadas para su posterior uso, permitiéndoles iniciar actividades específicas basadas en sus preferencias y objetivos. Esta característica adicional no solo añade una dimensión de personalización profunda, sino que también enriquece la utilidad de la aplicación en la vida diaria de los usuarios.

La implementación de Firebase como base de datos ha demostrado ser una elección acertada, ya que nos ha proporcionado una sólida capa de seguridad y autenticación. Las diversas funciones de seguridad que ofrece Firebase han permitido verificar la identidad de los usuarios y garantizar la integridad de la plataforma. Esta elección estratégica ha desempeñado un papel fundamental en la protección y estabilidad del sistema en su conjunto, brindando confianza tanto a nosotros como a los usuarios finales. Además, el correcto funcionamiento de almacenamiento de Firebase ha permitido gestionar eficientemente los datos, asegurando la disponibilidad y la confiabilidad de la información registrada por los usuarios.

Personalmente, este proyecto ha representado un período de crecimiento profesional verdaderamente invaluable para mí. Durante su desarrollo, no solo he adquirido nuevos conocimientos en lenguajes de programación, como JavaScript, sino que también he consolidado mis habilidades en HTML y CSS, elementos esenciales en el diseño web. Además, el descubrimiento y dominio del funcionamiento de Firebase como base de datos ha resultado fundamental, permitiéndonos almacenar de manera efectiva y segura los datos de los usuarios.

Una parte destacada de este proceso ha sido el aprovechamiento de herramientas avanzadas como Firebase y, en particular, el uso de Vuetify. Ya que ha brindado un gran valor a nuestro proyecto al proporcionar una serie de componentes visuales predefinidos que no solo agilizaron el desarrollo, sino que también mejoraron significativamente la estética de la aplicación.

Estas habilidades adquiridas y consolidadas no solo enriquecerán mi perfil profesional, sino que también serán activos fundamentales en mi futura carrera. La experiencia de llevar a cabo este proyecto desde cero, combinada con el aprendizaje profundo y práctico que he obtenido, me motiva a seguir innovando y mejorando la aplicación, así como a buscar continuamente oportunidades para crecer y prosperar en el campo del desarrollo de aplicaciones móviles y la tecnología en general. La emoción y el compromiso con los que abordé este proyecto desde cero han sido una fuente constante de motivación. Ver el resultado final, la culminación de esfuerzos, me impulsa a seguir perfeccionando la aplicación y a continuar evolucionando como profesional. Este proyecto ha sido un paso importante en mi trayectoria, y estoy ansiosa por

seguir contribuyendo a su mejora y expandiendo mis horizontes en el mundo del desarrollo de aplicaciones móviles.

M.8 LINEAS FUTURAS

Después de la implementación del proyecto de software, es valioso considerar una planificación hacia el futuro que abarque mejoras potenciales. Algunas ideas para potenciar el desarrollo de la aplicación incluyen:

- Incorporar un mapa detallado para optimizar la precisión en actividades como correr o andar en bicicleta. Esto permitiría una evaluación más completa al considerar tanto el tiempo como el esfuerzo físico invertido en cada ejercicio.
- Habilitar la conectividad con dispositivos electrónicos, posibilitando la monitorización de calorías quemadas y pasos dados a lo largo del día. Esta funcionalidad enriquecería la experiencia del usuario al proporcionar datos más precisos y en tiempo real.
- Enriquecer la plataforma con tutoriales en vídeo para la preparación de comidas saludables. Proporcionar orientación práctica respaldaría al usuario en la creación de platos nutritivos, promoviendo un enfoque integral hacia la salud.

Estos son ajustes que, con el avance de mi trayectoria profesional y la inversión más tiempo, aspiraría a implementar exitosamente. Mediante estos refinamientos, la aplicación podría aspirar a destacar aún más en el competitivo mercado, ofreciendo una propuesta aún más completa y atractiva para los usuarios.

M.9.REFERENCIAS

- Visual Studio Code. Recuperado el 01/09/2023 de [Visual Studio Code - Code Editing. Redefined](#)
- HTML Recuperado el 01/09/2023 de [HTML Tutorial \(w3schools.com\)](#)
- CSS. Recuperado el 01/09/2023 de [CSS Introduction \(w3schools.com\)](#)
- Lenguaje Javascript. Recuperado el 01/09/2023 de [JavaScript Tutorial \(w3schools.com\)](#)
- VueJS. Recuperado el 01/09/2023 de [Introduction | Vue.js \(vuejs.org\)](#)
- Vuetify. Recuperado el 01/09/2023 de [Why you should be using Vuetify — Vuetify \(vuetifyjs.com\)](#)
- Ionic. Recuperado el 01/09/2023 de [Ionic Framework - The Cross-Platform App Development Leader](#)
- npm: vue-material. Vue-Material. Recuperado el 01/09/2023 de <https://www.npmjs.com/package/vue-material>
- Nodejs. Recuperado el 01/09/2023 de [Acerca | Node.js \(nodejs.org\)](#)
- JSDoc. Recuperado el 01/09/2023 de [jsdoc-vuejs - npm \(npmjs.com\)](#)
- NPM. Recuperado el 01/09/2023 de [About npm | npm Docs \(npmjs.com\)](#)
- VisualParadigm. Recuperado el 01/09/2023 de [Ideal Modeling & Diagramming Tool for Agile Team Collaboration \(visual-paradigm.com\)](#)
- Firebase. Recuperado el 01/09/2023 de <https://firebase.google.com/docs/guides>
- Microsoft Project. Recuperado el 01/09/2023 de [Project Management Software | Microsoft Project](#)
- Dr. Francisco José García Peñalvo, Dra. María N. Moreno García, Dra. Alicia García Holgado y Dña. Andrea Vázquez Ingelmo. Tema 8 – UML. Unified Modeling Language. INGENIERÍA DE SOFTWARE I. Recuperado el 01/09/2023.
- Rodrigo Santamaría. Introducción. Sistemas Distribuidos. Recuperado el 01/09/2023.
- Rodrigo Santamaría. Introducción. Detalles de la asignatura. Recuperado el 01/09/2023.
- María N. Moreno García. Práctica 1- Estimación del esfuerzo. Gestión de Proyectos. Recuperado el 01/09/2023.
- María N. Moreno García y María Navarro Cáceres. Práctica 2 Planificación temporal. Gestión de Proyectos. Recuperado el 01/09/2023.