Logotipo

Descripción generada automáticamente

PLAN DE DIRECCIÓN DE PROYECTO

“FOODWISE”

Jordán Rosas. Ayax Rebolledo

Sección 004V – Profesor Gonzalo Paredes

2024

Contenido

[Índice de Ilustraciones 5](#_Toc181555787)

[Introducción 6](#_Toc181555788)

[Contexto del Proyecto 6](#_Toc181555789)

[Propósito del Plan 6](#_Toc181555790)

[Objetivos del Proyecto 7](#_Toc181555791)

[Alcance del Plan de Dirección 7](#_Toc181555792)

[Stakeholders del Proyecto 9](#_Toc181555793)

[Contexto 9](#_Toc181555794)

[Factores Clave del Proyecto 10](#_Toc181555795)

[Etapa preliminar Anteproyecto 11](#_Toc181555796)

[Evaluación de las necesidades 11](#_Toc181555797)

[Plan de Gestión de Beneficios 13](#_Toc181555798)

[Identificación de los beneficios 13](#_Toc181555799)

[Medición de los beneficios 14](#_Toc181555800)

[Responsabilidades 14](#_Toc181555801)

[Cronograma de realización de beneficios 14](#_Toc181555802)

[Riesgos asociados a los beneficios 15](#_Toc181555803)

[Mecanismos de revisión y control 15](#_Toc181555804)

[Sostenibilidad de los beneficios 15](#_Toc181555805)

[Metodología de gestión del proyecto 16](#_Toc181555806)

[Justificación del Uso de Metodología tradicional Waterfall 16](#_Toc181555807)

[Herramientas para la metodología Waterfall 16](#_Toc181555808)

[Desarrollo del plan de dirección del Proyecto 17](#_Toc181555809)

[Plan de gestión del alcance 17](#_Toc181555810)

[Propósito del plan de gestión del alcance 17](#_Toc181555811)

[Definición del alcance del proyecto 17](#_Toc181555812)

[Recopilación de Requisitos 18](#_Toc181555813)

[Listado de Requisitos funcionales 19](#_Toc181555814)

[Listado de Requisitos no funcionales 20](#_Toc181555815)

[Descripción del alcance del proyecto (Project Scope Statement) 20](#_Toc181555816)

[Criterios de Aceptación: 21](#_Toc181555817)

[Estructura de desglose del trabajo EDT 25](#_Toc181555818)

[Validación del alcance 25](#_Toc181555819)

[Control del alcance 28](#_Toc181555820)

[Cierre del Alcance 28](#_Toc181555821)

[Plan de gestión de los requerimientos 28](#_Toc181555822)

[Propósito del Plan de Gestión de Requerimientos 28](#_Toc181555823)

[Definición de los Requerimientos del Proyecto 29](#_Toc181555824)

[Listado de Requerimientos Funcionales 29](#_Toc181555825)

[Listado de Requerimientos No Funcionales 30](#_Toc181555826)

[Análisis de Dependencias 31](#_Toc181555827)

[Criterios de Aceptación 34](#_Toc181555828)

[Plan de gestión del cronograma 36](#_Toc181555829)

[Propósito de planificación del cronograma 36](#_Toc181555830)

[Definición y secuenciación de actividades 37](#_Toc181555831)

[Estimación de duración de las actividades 40](#_Toc181555832)

[Desarrollo del cronograma 42](#_Toc181555833)

[Control del cronograma 44](#_Toc181555834)

[Responsabilidades clave y gestión de cambios en el plan de gestión del cronograma 45](#_Toc181555835)

[Plan de gestión de Costos 46](#_Toc181555836)

[Propósito del plan de gestión de costos 46](#_Toc181555837)

[Estimación de Costos 46](#_Toc181555838)

[Presupuesto 46](#_Toc181555839)

[Control de Costos 46](#_Toc181555840)

[Indicadores Clave de Desempeño (KPI) 47](#_Toc181555841)

[Roles y responsabilidades 47](#_Toc181555842)

[Procesos de aprobación de cambios 47](#_Toc181555843)

[Herramientas y reportes financieros 48](#_Toc181555844)

[Plan de gestión de Calidad 48](#_Toc181555845)

[Plan de gestión de Riesgos 49](#_Toc181555846)

[Propósito del plan de riesgos 49](#_Toc181555847)

[Definición del plan de riesgos 49](#_Toc181555848)

[Desgloses de riesgos por área 49](#_Toc181555849)

[Estructura de desglose de riesgos RBS 51](#_Toc181555850)

[Plan de gestión de los Recursos 51](#_Toc181555851)

[Plan de gestión de Comunicaciones 51](#_Toc181555852)

[Plan de gestión de Adquisiciones 52](#_Toc181555853)

[Plan de gestión de Cambios 52](#_Toc181555854)

[Control integrado de cambios 52](#_Toc181555855)

[Control de costos y presupuestos 52](#_Toc181555856)

[Control de riesgos 52](#_Toc181555857)

[Control de recursos 52](#_Toc181555858)

[Control de Cronograma e Hitos 52](#_Toc181555859)

[Informe de estado de avance 52](#_Toc181555860)

[Estado Avance Costos y Presupuesto 52](#_Toc181555861)

[Estado Avance Recursos 52](#_Toc181555862)

[Estado Avance Cronograma e Hitos 52](#_Toc181555863)

[Acta de cierre del proyecto 52](#_Toc181555864)

# Índice de Ilustraciones

[Ilustración 1: Estructura de desglose del trabajo (EDT) 25](#_Toc181555773)

[Ilustración 2: Análisis de dependencias Registro de Usuarios y Seguridad 31](#_Toc181555774)

[Ilustración 3: Análisis de dependencias para llegar a generación de recomendación 34](#_Toc181555775)

[Ilustración 4: Estructura de desglose de riesgos RBS 51](#_Toc181555776)

# Introducción

El presente Plan de Dirección del Proyecto define el marco general para la gestión y ejecución del proyecto FoodWise, una aplicación móvil destinada a mejorar los hábitos alimenticios de los usuarios mediante la recolección y análisis de datos nutricionales. Este plan describe las metodologías, los procesos y los recursos que serán utilizados para garantizar que el proyecto se complete a tiempo, dentro del presupuesto y cumpliendo con los estándares de calidad establecidos.

## Contexto del Proyecto

En un contexto donde los malos hábitos alimenticios están asociados a un aumento de enfermedades crónicas, como la obesidad, la diabetes y los problemas cardiovasculares, surge la necesidad de herramientas que permitan a las personas gestionar su alimentación de manera más informada y consciente. El proyecto FoodWise tiene como objetivo cubrir esta necesidad mediante una aplicación que clasifica alimentos, ofrece información nutricional detallada y genera recomendaciones personalizadas a los usuarios para mejorar su salud a largo plazo.

## Propósito del Plan

El propósito de este Plan de Dirección del Proyecto es proporcionar un marco claro y estructurado para la gestión de todas las actividades relacionadas con el desarrollo de FoodWise. Este plan guía al equipo de proyecto a lo largo de todas las fases, desde la planificación inicial hasta la implementación y entrega final del producto. Además, establece las bases para la gestión del cronograma, los recursos, los riesgos y la comunicación con y entre los stakeholders.

## Objetivos del Proyecto

El principal objetivo de FoodWise es desarrollar una plataforma móvil intuitiva que permita a los usuarios:

* Ingresar manualmente la información nutricional de los alimentos consumidos.
* Clasificar los alimentos en diferentes categorías.
* Recibir recomendaciones nutricionales basadas en patrones de consumo y en las necesidades de salud del usuario.
* Ofrecer alertas y sugerencias personalizadas para prevenir el consumo excesivo de nutrientes perjudiciales para la salud.

Adicionalmente, FoodWise busca fomentar una alimentación saludable y consciente, mejorando la calidad de vida de sus usuarios.

## Alcance del Plan de Dirección

El Plan de Dirección del Proyecto FoodWise cubre todas las áreas clave de la gestión de proyectos, asegurando que se cumplan los objetivos establecidos y se gestionen adecuadamente los recursos y riesgos involucrados. A continuación, se detalla cada una de las áreas cubiertas por este plan:

* Plan de Gestión del Alcance: Define los límites del proyecto y los entregables esperados y no esperados, garantizando que el proyecto no se expanda fuera de los objetivos establecidos sin la debida autorización.
* Plan de Gestión de los Requerimientos: Establece los procedimientos para la recolección, documentación y validación de los requerimientos del proyecto, asegurando que todas las necesidades de los stakeholders sean consideradas.
* Plan de Gestión del Cronograma: Define la metodología y las herramientas utilizadas para desarrollar y gestionar el cronograma del proyecto, asegurando que las tareas se completen dentro de los plazos establecidos.
* Plan de Gestión de Costos: Establece los procedimientos para estimar, presupuestar y controlar los costos del proyecto, asegurando que se mantenga dentro del presupuesto asignado.
* Plan de Gestión de Calidad: Establece los estándares de calidad que se deben cumplir en el desarrollo del proyecto y define los procedimientos para garantizar que estos estándares sean alcanzados.
* Plan de Gestión de los Recursos: Define cómo se asignarán y gestionarán los recursos humanos y materiales necesarios para completar el proyecto dentro de los plazos y el presupuesto.
* Plan de Gestión de Comunicaciones: Establece los canales, la frecuencia y los métodos de comunicación entre el equipo del proyecto y los stakeholders, asegurando un flujo de información constante y claro.
* Plan de Gestión de Riesgos: Define los procesos para identificar, analizar, mitigar y monitorear los riesgos del proyecto, asegurando que los impactos negativos sean minimizados.
* Plan de Gestión de Adquisiciones: Describe los procedimientos para la adquisición de recursos externos necesarios para el proyecto, como servicios o productos, y cómo se gestionarán las relaciones con los proveedores.
* Plan de Gestión de Cambios: Establece los procedimientos para gestionar los cambios en el proyecto, asegurando que cualquier modificación al alcance, cronograma o presupuesto sea adecuadamente analizada y aprobada.
* Control Integrado de Cambios: Define los mecanismos para coordinar los cambios en el proyecto, asegurando que los cambios en un área no afecten negativamente a otras, siempre cuidando las líneas base del proyecto.
* Control de Costos y Presupuestos: Establece los procesos para monitorear y controlar los costos durante la ejecución del proyecto, garantizando que el proyecto se mantenga dentro del presupuesto asignado.
* Control de Riesgos: Define los procedimientos para el monitoreo continuo de los riesgos del proyecto y la implementación de estrategias de mitigación cuando sea necesario.
* Control de Recursos: Establece los procedimientos para asegurar que los recursos humanos y materiales se utilicen de manera eficiente y que se ajusten según sea necesario durante el ciclo de vida del proyecto.
* Control del Cronograma e Hitos: Define los procedimientos para el monitoreo y control del cronograma, asegurando que se respeten los plazos establecidos y que los hitos clave del proyecto se cumplan de manera oportuna. Para la gestión del proyecto, se ha adoptado el enfoque tradicional mediante la metodología Waterfall, lo cual permite un proceso secuencial y estructurado. El proyecto se dividirá en seis fases, cada una con entregables claramente definidos y revisiones al término de cada fase para asegurar la calidad y el cumplimiento de los requisitos.
* Cada fase del proyecto (incluyendo los requisitos, diseño, implementación, pruebas, implementación final y mantenimiento) será documentada y revisada antes de pasar a la siguiente, lo que garantiza un progreso sólido y controlado hacia los objetivos definidos de FoodWise.

## Stakeholders del Proyecto

Los stakeholders clave en este proyecto incluyen:

Equipo de Desarrollo: Compuesto por Ayax Rebolledo y Jordán Rosas, quienes son los principales responsables del desarrollo técnico.

Stakeholder Académico: Profesor Gonzalo Alejandro Paredes Recabarren, quien supervisa el desarrollo del proyecto y proporciona orientación clave.

Usuarios Finales: Familias e individuos interesados en mejorar sus hábitos alimenticios.

# Contexto

En la actualidad, la alimentación inadecuada y los malos hábitos alimenticios se han convertido en un problema crítico a nivel global, contribuyendo al aumento de enfermedades crónicas como la obesidad, la diabetes, y problemas cardiovasculares. Muchas personas carecen de la información necesaria para gestionar su nutrición de manera efectiva, lo que impacta negativamente en su salud y bienestar a largo plazo. En este contexto, el proyecto FoodWise nace como una respuesta tecnológica para ayudar a los usuarios a comprender mejor sus patrones alimenticios, mejorar sus decisiones nutricionales y prevenir posibles problemas de salud relacionados con una mala alimentación.

FoodWise es una aplicación móvil diseñada para proporcionar a los usuarios información detallada sobre los alimentos que consumen, permitiéndoles ingresar manualmente datos nutricionales y recibir recomendaciones personalizadas basadas en sus hábitos y necesidades de salud. La aplicación se propone como una herramienta de apoyo para individuos y familias que buscan mejorar su calidad de vida mediante una alimentación más consciente y equilibrada.

Este proyecto se desarrolla en un contexto en el que la tecnología está cada vez más integrada en la vida diaria de las personas, convirtiendo los dispositivos móviles en una extensión de su capacidad para acceder a información clave y tomar decisiones informadas. FoodWise pretende aprovechar esta tendencia para facilitar la gestión de la alimentación, utilizando técnicas avanzadas como la minería de datos y algoritmos de recomendación para ofrecer a los usuarios una experiencia personalizada y enriquecedora.

## Factores Clave del Proyecto

* Creciente preocupación por la salud: La sociedad está cada vez más preocupada por los efectos de la mala alimentación, lo que crea una oportunidad para soluciones tecnológicas que promuevan hábitos más saludables.
* Proliferación de dispositivos móviles: La adopción generalizada de smartphones y otras tecnologías móviles permite el desarrollo de aplicaciones que los usuarios pueden integrar fácilmente en su vida diaria.
* Minería de datos y personalización: El análisis de grandes volúmenes de datos y la capacidad de generar recomendaciones personalizadas son elementos clave que añaden valor a la aplicación, diferenciándola de otros productos similares en el mercado.
* Enfoque en la prevención de enfermedades: A través de alertas y recomendaciones, FoodWise busca no solo mejorar los hábitos alimenticios, sino también prevenir enfermedades asociadas con el consumo excesivo de ciertos nutrientes, como sodio, azúcares y grasas.

# Etapa preliminar Anteproyecto

## Evaluación de las necesidades

1. Necesidad de Mejorar los Hábitos Alimenticios

La alimentación deficiente, caracterizada por el exceso de alimentos ultra procesados, ricos en grasas saturadas, sodio y azúcares, se ha convertido en un factor determinante en la proliferación de enfermedades crónicas no transmisibles, como la obesidad, la diabetes y los trastornos cardiovasculares. Esto afecta a personas de todas las edades, pero especialmente a adultos entre 18 y 55 años, quienes son los principales responsables de la compra y planificación de alimentos.

A pesar de la creciente conciencia sobre la relación entre la alimentación y la salud, muchas personas carecen de las herramientas necesarias para gestionar su nutrición de manera adecuada. Los consumidores enfrentan dificultades para entender las etiquetas nutricionales y, a menudo, no cuentan con una forma clara de medir su ingesta de nutrientes a lo largo del tiempo. (Foodwise, 2024)

2. Falta de Herramientas Tecnológicas para la Gestión Nutricional Personalizada

Existen pocas herramientas accesibles que permitan a los usuarios gestionar sus hábitos alimenticios de manera proactiva, personalizada y aún más importante, en tiempo real. Si bien hay aplicaciones que proporcionan información nutricional básica, pocas ofrecen recomendaciones personalizadas basadas en patrones de consumo, necesidades de salud específicas y metas nutricionales. Los usuarios necesitan una plataforma que les permita:

* Ingresar fácilmente información nutricional de los alimentos que consumen, ya sean envasados o frescos.
* Clasificar los alimentos de acuerdo a categorías como lácteos, frutas, verduras, carnes, etc.
* Recibir alertas y recomendaciones sobre el consumo de ciertos nutrientes, sugiriendo la reducción o aumento de alimentos según las necesidades personales y los objetivos de salud.
* Prevenir problemas de salud relacionados con una alimentación desequilibrada.

3. Falta de Información Nutrimental de Alimentos No Etiquetados

Un desafío importante es que muchos alimentos frescos o platos preparados no cuentan con información nutricional clara. Los usuarios necesitan una manera simple de acceder a estos datos, ya sea ingresándolos manualmente o estimando los nutrientes de platos comunes a través de bases de datos confiables.

4. Necesidad de Recomendaciones Personalizadas Basadas en Minería de Datos

Dado que cada persona tiene necesidades nutricionales diferentes, influenciadas por factores como la edad, el nivel de actividad física, y condiciones de salud particulares (diabetes, hipertensión, alergias, etc.), es necesario contar con un sistema que pueda procesar datos de consumo y generar recomendaciones personalizadas. Los usuarios requieren una aplicación que, mediante minería de datos y algoritmos de recomendación, pueda:

* Identificar patrones de consumo y proporcionar alertas sobre el exceso o deficiencia de nutrientes.
* Sugerir metas nutricionales específicas que ayuden a mejorar su salud general.
* Anticipar posibles deficiencias o excesos nutricionales, alertando a los usuarios antes de que se desarrollen problemas de salud.

5. Educación Nutricional y Prevención de Enfermedades

Las personas requieren más que información sobre calorías o nutrientes aislados; necesitan educación nutricional que les ayude a entender el impacto de sus decisiones alimenticias en su bienestar general. Además, muchos usuarios no son conscientes de cómo sus hábitos alimenticios pueden contribuir al desarrollo de enfermedades crónicas. La aplicación debe ofrecerles información sobre:

* Los riesgos de consumir en exceso ciertos alimentos (altos en azúcares, sodio, grasas saturadas).
* Los beneficios del consumo de alimentos equilibrados (ricas en fibra, proteínas magras, vitaminas y minerales).
* Consejos para alcanzar una dieta equilibrada, basados en sus patrones actuales de consumo.

6. Necesidad de Simplicidad y Usabilidad en las Aplicaciones Nutricionales

Muchos usuarios encuentran complicado el uso de aplicaciones actuales que requieren procesos tediosos para registrar los alimentos. Existe una necesidad clara de una interfaz amigable y eficiente que permita el registro rápido y fácil de los alimentos consumidos. Esto garantizará una mayor adherencia y uso continuo de la aplicación, maximizando su impacto positivo en los hábitos alimenticios.

# Plan de Gestión de Beneficios

## Identificación de los beneficios

El proyecto FoodWise tiene como objetivo generar los siguientes beneficios principales:

* Mejora de los hábitos alimenticios: A través de recomendaciones personalizadas basadas en la ingesta de alimentos y el análisis de patrones de consumo.
* Conciencia sobre la salud: Los usuarios desarrollarán una mayor conciencia sobre su ingesta de nutrientes, lo que puede prevenir enfermedades crónicas como la obesidad y la diabetes.
* Facilidad en la gestión de la nutrición diaria: Proporcionar a los usuarios una herramienta eficiente y fácil de usar para monitorear su dieta y recibir alertas sobre excesos o deficiencias de nutrientes.
* Acceso a información personalizada: La aplicación permitirá a los usuarios obtener información específica sobre su consumo de alimentos, ajustada a sus necesidades individuales.

## Medición de los beneficios

Para garantizar que los beneficios se estén logrando, se utilizarán las siguientes métricas:

* Nivel de adopción de la aplicación: Medir la cantidad de usuarios que descargan, registran y utilizan la aplicación de manera regular.
* Mejora en los patrones alimenticios: Realizar encuestas o análisis de datos dentro de la aplicación para identificar cambios positivos en la dieta de los usuarios (p. ej., aumento en el consumo de alimentos saludables o reducción de alimentos altos en sodio y grasas).
* Satisfacción de los usuarios: Realizar encuestas periódicas para medir el grado de satisfacción de los usuarios con las recomendaciones nutricionales y la facilidad de uso de la aplicación.
* Reducción de hábitos nocivos: Analizar el impacto de las alertas y recomendaciones sobre el consumo excesivo de ciertos nutrientes (p. ej., disminución del consumo de alimentos con alto contenido de azúcar).

## Responsabilidades

Dentro del equipo se deben asignar roles y responsabilidades para garantizar la gestión de este plan:

* Equipo de desarrollo será el responsable de garantizar la funcionalidad de la aplicación, alineándose con la generación de los beneficios esperados.
* La calidad de las recomendaciones entregadas por la aplicación debe ser asegurada por entrevistas y feedback con profesionales nutricionistas y del área de la salud.
* Debe existir un seguimiento para maximizar la adopción de la aplicación por parte de los usuarios, y se debe analizar de forma continua el feddback de estos para futuras mejoras continuas.

## Cronograma de realización de beneficios

* Corto plazo (primeros 3 meses): Beneficios como la facilidad de uso de la aplicación y la mejora en la conciencia de los usuarios sobre sus patrones de consumo deberían ser evidentes desde las primeras semanas de uso.
* Mediano plazo (6 meses): Los usuarios deberían comenzar a ver cambios más significativos en sus hábitos alimenticios y mejoras en su salud general, respaldadas por un uso continuo de la aplicación y las alertas personalizadas.
* Largo plazo (12 meses o más): La prevención de enfermedades crónicas y la creación de hábitos alimenticios sostenibles serán beneficios a largo plazo, medidos a través de encuestas de salud y datos recogidos en la aplicación.

## Riesgos asociados a los beneficios

* Baja adopción de la aplicación: Los usuarios podrían no adoptarla debido a la falta de interés o desconocimiento sobre los beneficios que ofrece.
* Resistencia a cambiar hábitos: Los usuarios podrían no seguir las recomendaciones o no cambiar sus hábitos alimenticios, lo que limitaría el impacto del proyecto.
* Recomendaciones imprecisas: Si los algoritmos de minería de datos no proporcionan recomendaciones precisas o relevantes, esto podría impactar la satisfacción de los usuarios y la efectividad del proyecto.
* Complejidad en el uso: Si la interfaz de la aplicación no es lo suficientemente intuitiva, podría desincentivar su uso continuo.

## Mecanismos de revisión y control

* Informes periódicos: Se realizarán informes trimestrales para evaluar el progreso en la adopción de la aplicación y la efectividad de las recomendaciones personalizadas.
* Revisión de encuestas de usuarios: Se llevará a cabo una revisión constante de los datos recopilados a través de las encuestas de satisfacción y el análisis de los patrones de consumo de los usuarios.
* Análisis de datos en tiempo real: Se utilizarán paneles de control y herramientas de análisis de datos para monitorear el uso de la aplicación y el impacto de las recomendaciones nutricionales en los usuarios.

## Sostenibilidad de los beneficios

* Actualización continua: El equipo de FoodWise se comprometerá a realizar actualizaciones regulares para mejorar la funcionalidad de la aplicación y mantener las recomendaciones actualizadas con las últimas investigaciones en nutrición. Estas actualizaciones van de las manos con futuras mejoras que expandan en alcance de la aplicación, como por ejemplo el uso de dispositivos móviles (smartwatch) que monitoreen la actividad física de los usuarios.
* Monitoreo a largo plazo: Los beneficios serán monitoreados de manera continua a través de análisis de datos de los usuarios, asegurando que los hábitos alimenticios mejorados se mantengan con el tiempo.
* Soporte y comunidad: La creación de una comunidad de usuarios y el soporte constante ayudarán a los usuarios a mantener los beneficios a largo plazo, fomentando la interacción y el intercambio de experiencias en torno a una alimentación más saludable.

# Metodología de gestión del proyecto

Para el desarrollo de FoodWise, se utilizará la metodología Waterfall. Este enfoque de trabajo es secuencial y predictivo, lo que implica que cada fase del proyecto debe completarse antes de iniciar la siguiente. Las fases incluyen: requerimientos, diseño, implementación, pruebas, implementación final y mantenimiento. Este enfoque permite una documentación detallada y una planificación anticipada que garantizan una estructura sólida y un control riguroso en cada etapa.

## Justificación del Uso de Metodología tradicional Waterfall

La naturaleza secuencial de Waterfall asegura que cada etapa se documente con precisión, lo cual facilita el control y seguimiento del progreso. En el caso de FoodWise, la documentación completa de requisitos y diseño es esencial para evitar cambios en etapas avanzadas, asegurando que el producto final esté alineado con las expectativas iniciales. FoodWise requiere una estructura definida desde el inicio, especialmente en aspectos como la base de datos y la interfaz de usuario. Como la funcionalidad deseada está bien delimitada y las necesidades de los stakeholders son claras, Waterfall permite prever el flujo del proyecto sin necesidad de cambios constantes.

Por otro lado, dado que cada fase en Waterfall depende del cumplimiento de la anterior, la metodología reduce los riesgos asociados con ajustes de último minuto. Esto es ideal para FoodWise, ya que permite que cada fase sea revisada y aprobada antes de avanzar, asegurando la calidad de cada componente.

### Herramientas para la metodología Waterfall

Microsoft Word para crear y manejar la documentación del proyecto, principalmente el plan de dirección de proyecto.

Para la gestión del cronograma, estamos utilizando Monday, y su tablero para para organizar fases y subtareas, permitiendo un seguimiento claro y visual de las dependencias y progreso.

Respecto a las herramientas de diseño del sistema y prototipado, usaremos Figma para la interfaz de usuario y prototipos de Foodwise.

Utilizaremos Lucidchart para los diagramas de flujo, arquitectura del sistema, y esquema de base de datos, permitiendo visualizar la estructura y funcionamiento del sistema en cada fase.

# Desarrollo del plan de dirección del Proyecto

## Plan de gestión del alcance

### Propósito del plan de gestión del alcance

El propósito del Plan de Gestión del Alcance es proporcionar una guía detallada sobre cómo se define, documenta, valida y controla el alcance del proyecto FoodWise. Este plan asegura que el proyecto incluya todo el trabajo necesario para cumplir con los objetivos del proyecto y que no se realicen tareas fuera del alcance aprobado.

### Definición del alcance del proyecto

El proyecto FoodWise tiene como objetivo el desarrollo de una aplicación móvil que permita a los usuarios gestionar su alimentación mediante el registro manual de información nutricional, la generación de recomendaciones personalizadas, y el monitoreo de hábitos alimenticios. La aplicación también incluirá soporte para restricciones alimentarias y ofrecerá recomendaciones basadas en un modelo de minería de datos. Para cumplir estos objetivos de manera eficiente, FoodWise implementará una arquitectura de tres capas:

Capa de Presentación (Frontend): La aplicación móvil será desarrollada en React Native y estará disponible para iOS y Android. Esta capa gestionará la interacción directa con el usuario, proporcionando pantallas intuitivas para el ingreso de datos nutricionales, visualización de recomendaciones, configuración de alertas, y visualización de patrones de consumo.

Capa de Lógica de Negocio (Backend): Usando Node.js o Django, esta capa procesará todas las reglas de negocio y el flujo de datos. Incluirá el API REST que facilita la comunicación entre la app y el backend, un módulo de recomendaciones que utiliza un modelo de árbol de decisiones para generar alertas y sugerencias personalizadas, y un motor de minería de datos y predicción que analiza los hábitos de consumo y los patrones históricos.

Capa de Datos (Base de Datos): La información será almacenada en una base de datos PostgreSQL o MongoDB. Esta capa gestionará los datos nutricionales, los perfiles de usuario, el historial de consumo y las configuraciones de alertas. De este modo, la aplicación puede realizar consultas rápidas y eficaces para personalizar las recomendaciones y generar alertas nutricionales basadas en los datos del usuario.

Los entregables definidos son:

* Aplicación móvil funcional para iOS y Android.
* Sistema de recomendaciones personalizadas basado en un modelo de árbol de decisiones.
* Gestión de perfiles personalizados de usuarios.
* Generación de alertas nutricionales y de consumo.
* Recomendaciones para mejorar hábitos alimenticios.

Límites del proyecto:

* No se incluirá un sistema de descomposición automática de recetas.
* No se incluirá el análisis de comidas preparadas en tiempo real.

### Recopilación de Requisitos

Identificamos las necesidades y expectativas de los stakeholders mediante la recopilación de datos para documentar los requisitos funcionales y no funcionales del proyecto. Las técnicas de recopilación incluyen:

* Entrevistas y/o cuestionario con stakeholders clave (profesor Gonzalo Alejandro Paredes Recabarren, usuarios y profesional nutricionista Belen Isuani).
* Revisiones de normativa alimentaria.
* Encuestas a usuarios potenciales para identificar las expectativas sobre la funcionalidad de la aplicación.
* Análisis técnico para definir los recursos y herramientas necesarios para el desarrollo.
* Entrevistas técnicas e investigación con el equipo de desarrollo y profesor guía para identificar las necesidades de infraestructura y tecnología.
* Estudio de factibilidad técnica para asegurar que la tecnología seleccionada es adecuada para los objetivos del proyecto.

### Listado de Requisitos funcionales

|  |  |  |  |
| --- | --- | --- | --- |
| **N°** | **ID** | **Requerimiento** | **Tipo de Requerimiento** |
| 1 | RF-1 | Ingreso manual de datos nutricionales de alimentos | Funcional |
| 2 | RF-2 | Clasificación de alimentos por categorías | Funcional |
| 3 | RF-3 | Creación de perfiles personalizados con metas nutricionales | Funcional |
| 4 | RF-4 | Generación de recomendaciones personalizadas basadas en patrones de consumo | Funcional |
| 5 | RF-5 | Proyecciones de bienestar y salud a largo plazo | Funcional |
| 6 | RF-6 | Soporte para restricciones alimentarias (diabetes, celíacos, etc.) | Funcional |
| 7 | RF-7 | Generación de alertas dinámicas para prevenir el consumo excesivo de nutrientes | Funcional |
| 8 | RF-8 | Diseño de interfaz intuitiva y fácil de usar | Funcional |
| 9 | RF-9 | Compatibilidad con gestos táctiles para navegación | Funcional |
| 10 | RF-10 | Soporte para diferentes resoluciones de pantalla | Funcional |
| 11 | RF-11 | Registro de usuarios con correo electrónico y contraseña | Funcional |
| 12 | RF-12 | Posibilidad de registro mediante cuentas de terceros (Google, Facebook, etc.) | Funcional |
| 13 | RF-13 | Recuperación de contraseña mediante correo electrónico | Funcional |
| 14 | RF-14 | Generación de alertas sobre exceso de consumo de nutrientes (sodio, grasas, azúcares) | Funcional |
| 15 | RF-15 | Notificaciones personalizadas para recordar la ingesta de alimentos o nutrientes específicos | Funcional |
| 16 | RF-16 | Implementación de un modelo de árbol de decisiones para generar recomendaciones personalizadas | Funcional |
| 17 | RF-17 | El modelo debe analizar patrones de consumo y hábitos del usuario para sugerir cambios alimenticios | Funcional |

### Listado de Requisitos no funcionales

|  |  |  |  |
| --- | --- | --- | --- |
| **N°** | **ID** | **Requerimiento** | **Tipo de Requerimiento** |
| 1 | RNF 1 | La aplicación debe ser multiplataforma (iOS y Android) | No Funcional |
| 2 | RNF 2 | Escalabilidad para soportar el crecimiento de usuarios | No Funcional |
| 3 | RNF 3 | Seguridad en el manejo de datos personales y nutricionales | No Funcional |
| 4 | RNF 4 | Actualizaciones frecuentes sin interrupciones del servicio | No Funcional |
| 5 | RNF 5 | Optimización de rendimiento (carga rápida y fluidez en el uso) | No Funcional |
| 6 | RNF 6 | Compatibilidad con normativas alimentarias y de privacidad | No Funcional |
| 7 | RNF 7 | La interfaz debe ser responsiva y adaptarse a distintos tamaños de dispositivos | No Funcional |
| 8 | RNF 8 | Cifrado de contraseñas y protección de datos del usuario durante el registro | No Funcional |
| 9 | RNF 9 | Las alertas deben configurarse para mostrarse en horarios específicos elegidos por el usuario | No Funcional |
| 10 | RNF 10 | Las alertas deben ser precisas y no interrumpir el funcionamiento de otras funciones de la aplicación | No Funcional |
| 11 | RNF 11 | El modelo debe ser optimizado para que las decisiones se tomen en tiempo real sin afectar el rendimiento | No Funcional |

### Descripción del alcance del proyecto (Project Scope Statement)

Se incluye una descripción detallada de los principales entregables, los criterios de aceptación, las exclusiones del proyecto, las restricciones y las asunciones clave.

Entregables del Proyecto:

* Aplicación Móvil: Desarrollo de una app multiplataforma para iOS y Android.
* Interfaz de usuario intuitiva y personalizable.
* Base de Datos Nutricional: Almacenamiento de información nutricional de alimentos ingresada manualmente por el usuario.
* Soporte para la clasificación de alimentos en categorías como frutas, verduras, lácteos, etc.
* Modelo de Árbol de Decisiones: Generación de recomendaciones personalizadas basadas en patrones de consumo y hábitos alimenticios.
* Proyecciones de bienestar a largo plazo.
* Gestión de Perfiles Personalizados: Creación de perfiles con restricciones alimentarias y metas personalizadas.
* Soporte para múltiples perfiles por familia.

### Criterios de Aceptación:

|  |  |  |  |
| --- | --- | --- | --- |
| **REQUERIMIENTOS FUNCIONALES** | | | |
| **N°** | **ID** | **Requerimiento** | **CRITERIOS DE ACEPTACIÓN** |
| 1 | RF-1 | Ingreso manual de datos nutricionales de alimentos | El usuario debe poder ingresar manualmente la información nutricional de al menos 10 alimentos en una sesión sin errores. |
| 2 | RF-2 | Clasificación de alimentos por categorías | Los alimentos deben poder clasificarse correctamente en las categorías definidas y deben aparecer correctamente en la interfaz y en la base de datos. |
| 3 | RF-3 | Creación de perfiles personalizados con metas nutricionales | El usuario debe poder crear, editar y guardar un perfil personalizado, que debe ser accesible en sesiones posteriores. |
| 4 | RF-4 | Generación de recomendaciones personalizadas basadas en patrones de consumo | El sistema debe generar recomendaciones basadas en los patrones de consumo del usuario dentro de un plazo de 2 segundos. |
| 5 | RF-5 | Proyecciones de bienestar y salud a largo plazo | El sistema debe proyectar los impactos en la salud del usuario basados en el consumo registrado y sugerir mejoras nutricionales. |
| 6 | RF-6 | Soporte para restricciones alimentarias (diabetes, celíacos, etc.) | El usuario debe poder seleccionar restricciones alimentarias en su perfil y el sistema debe generar alertas sobre alimentos no recomendados. |
| 7 | RF-7 | Generación de alertas dinámicas para prevenir el consumo excesivo de nutrientes | El sistema debe generar alertas dinámicas en tiempo real si el usuario excede el consumo de ciertos nutrientes, reflejándose en la interfaz. |
| 8 | RF-8 | Diseño de interfaz intuitiva y fácil de usar | La interfaz debe ser fácil de usar, con navegación intuitiva y accesible para todos los usuarios. |
| 9 | RF-9 | Compatibilidad con gestos táctiles para navegación | La navegación mediante gestos debe funcionar sin errores en todas las pantallas de la aplicación. |
| 10 | RF-10 | Soporte para diferentes resoluciones de pantalla | La aplicación debe funcionar correctamente en dispositivos con distintas resoluciones de pantalla sin distorsión visual. |
| 11 | RF-11 | Registro de usuarios con correo electrónico y contraseña | El usuario debe poder registrarse utilizando correo electrónico y contraseña, con validación de los datos ingresados. |
| 12 | RF-12 | Posibilidad de registro mediante cuentas de terceros (Google, Facebook, etc.) | El usuario debe poder registrarse mediante cuentas de Google o Facebook, y los datos deben sincronizarse correctamente. |
| 13 | RF-13 | Recuperación de contraseña mediante correo electrónico | El usuario debe poder recuperar su contraseña mediante el envío de un enlace de recuperación al correo electrónico. |
| 14 | RF-14 | Generación de alertas sobre exceso de consumo de nutrientes (sodio, grasas, azúcares) | El sistema debe generar alertas si el usuario excede los límites diarios recomendados de sodio, grasas o azúcares. |
| 15 | RF-15 | Notificaciones personalizadas para recordar la ingesta de alimentos o nutrientes específicos | El usuario debe recibir notificaciones personalizadas para recordar la ingesta de ciertos alimentos o nutrientes, según sus hábitos alimenticios. |
| 16 | RF-16 | Implementación de un modelo de árbol de decisiones para generar recomendaciones personalizadas | El modelo de árbol de decisiones debe generar recomendaciones personalizadas en base a los hábitos alimenticios registrados por el usuario. |
| 17 | RF-17 | El modelo debe analizar patrones de consumo y hábitos del usuario para sugerir cambios alimenticios | El modelo debe analizar los patrones de consumo del usuario y sugerir cambios alimenticios de acuerdo a los hábitos previos. |
| **REQUERIMIENTOS NO FUNCIONALES** | | | |
| **N°** | **ID** | **Requerimiento** | **CRITERIOS DE ACEPTACIÓN** |
| 1 | RNF 1 | La aplicación debe ser multiplataforma (iOS y Android) | La aplicación debe ser completamente funcional en dispositivos iOS y Android, con una interfaz coherente y un rendimiento óptimo. |
| 2 | RNF 2 | Escalabilidad para soportar el crecimiento de usuarios | El sistema debe escalar para soportar al menos 10,000 usuarios simultáneos sin afectar el rendimiento. |
| 3 | RNF 3 | Seguridad en el manejo de datos personales y nutricionales | Todos los datos personales y nutricionales ingresados por el usuario deben estar cifrados y protegidos con autenticación segura. |
| 4 | RNF 4 | Actualizaciones frecuentes sin interrupciones del servicio | La aplicación debe permitir actualizaciones frecuentes sin interrumpir la experiencia del usuario o el funcionamiento del sistema. |
| 5 | RNF 5 | Optimización de rendimiento (carga rápida y fluidez en el uso) | El tiempo de carga de cada pantalla principal de la aplicación no debe exceder los 3 segundos. |
| 6 | RNF 6 | Compatibilidad con normativas alimentarias y de privacidad | El sistema debe cumplir con las normativas alimentarias y de protección de datos vigentes en la región. |
| 7 | RNF 7 | La interfaz debe ser responsiva y adaptarse a distintos tamaños de dispositivos | La interfaz debe ser completamente responsiva y adaptarse a diferentes tamaños de pantalla sin pérdida de funcionalidad. |
| 8 | RNF 8 | Cifrado de contraseñas y protección de datos del usuario durante el registro | Las contraseñas y los datos del usuario deben estar cifrados y protegidos durante el proceso de registro. |
| 9 | RNF 9 | Las alertas deben configurarse para mostrarse en horarios específicos elegidos por el usuario | El sistema debe permitir configurar alertas que se muestren en horarios específicos seleccionados por el usuario. |
| 10 | RNF 10 | Las alertas deben ser precisas y no interrumpir el funcionamiento de otras funciones de la aplicación | Las alertas deben ser precisas y mostrarse sin interrumpir otras funcionalidades críticas de la aplicación. |
| 11 | RNF 11 | El modelo debe ser optimizado para que las decisiones se tomen en tiempo real sin afectar el rendimiento | El modelo de árbol de decisiones debe tomar decisiones en tiempo real sin impactar el rendimiento general de la aplicación. |

Exclusiones del Alcance:

No se incluirá la descomposición automática de recetas ni el análisis de alimentos complejos como comidas preparadas.

Restricciones del Proyecto:

* El desarrollo del proyecto debe completarse en un plazo de 18 semanas según el cronograma.
* Los recursos disponibles son limitados a dos personas en el equipo de desarrollo.

Asunciones Clave:

* Los usuarios podrán ingresar manualmente los datos nutricionales de los alimentos.
* Se asumirán hábitos alimenticios con base en los registros previos de consumo.

### Estructura de desglose del trabajo EDT

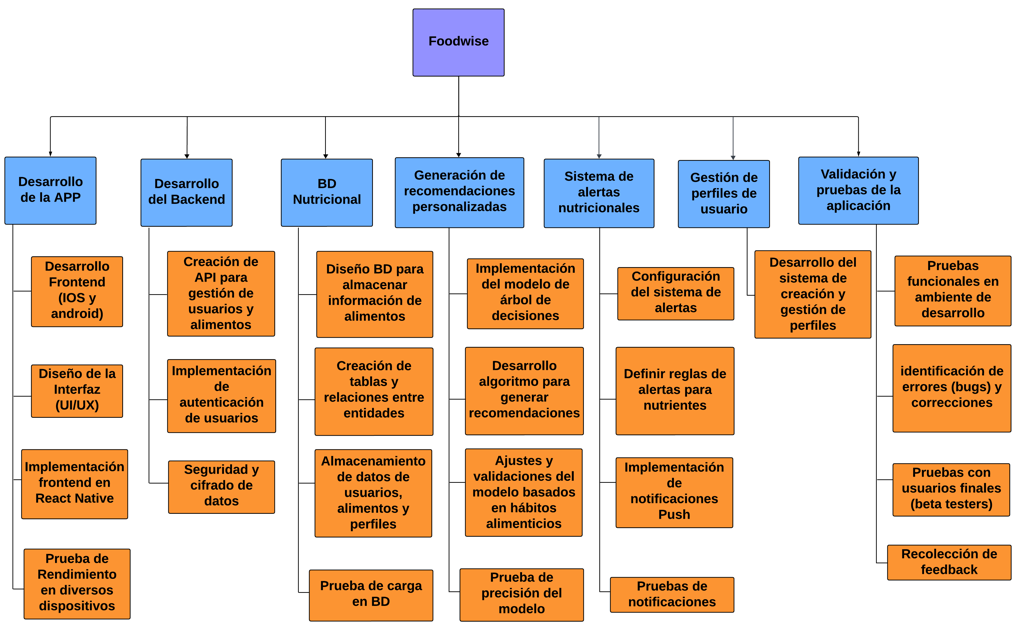


Ilustración 1: Estructura de desglose del trabajo (EDT)

### Validación del alcance

Para la obtención de aceptación formal de los entregables del proyecto, se deben realizar las siguientes actividades para validar el alcance:

* Revisiones con stakeholders clave: Se realizarán revisiones periódicas con los stakeholders, incluidos los usuarios potenciales y el profesor, para asegurar que los entregables cumplan con los requisitos definidos.
* Pruebas de aceptación: Los entregables se validarán a través de pruebas con usuarios y stakeholders para garantizar que cumplan con los criterios de aceptación.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| N° | **Nombre de la Prueba** | **Objetivo** | **Criterio de Aceptación** | **Porcentaje de Aceptación (%)** |
| 1 | Prueba de Registro de Usuarios | Verificar que los usuarios puedan registrarse con correo electrónico y contraseña sin errores. | El sistema debe permitir el registro exitoso, almacenar los datos y enviar un correo de confirmación. | 90% - 100% |
| 2 | Prueba de Inicio de Sesión | Comprobar que los usuarios puedan iniciar sesión con sus credenciales. | El sistema debe permitir iniciar sesión con credenciales válidas y mostrar mensajes de error claros para credenciales incorrectas. | 90% - 100% |
| 3 | Prueba de Diseño de la Interfaz | Verificar que la interfaz de usuario sea intuitiva y fácil de usar. | La interfaz debe cargar correctamente en dispositivos móviles y permitir la navegación fluida entre secciones. | 90% - 100% |
| 4 | Pruebas de Multiplataforma (iOS/Android) | Validar que la aplicación funcione de manera consistente en ambas plataformas. | La interfaz debe mantenerse coherente y funcional en dispositivos iOS y Android. | 90% - 100% |
| 5 | Prueba de Almacenamiento de Datos | Verificar que los datos nutricionales se almacenen correctamente en la base de datos. | El sistema debe guardar y recuperar la información nutricional en menos de 1 segundo. | 90% - 100% |
| 6 | Prueba de Seguridad de Datos | Comprobar que los datos de los usuarios están protegidos. | Los datos personales y nutricionales deben estar cifrados y la autenticación debe ser segura. | 90% - 100% |
| 7 | Prueba de Sincronización de Perfiles | Validar que los perfiles se sincronicen correctamente entre dispositivos. | El usuario debe poder iniciar sesión en distintos dispositivos y encontrar su perfil actualizado sin pérdida de datos. | 90% - 100% |
| 8 | Prueba de Precisión de Recomendaciones | Comprobar que el sistema genera recomendaciones precisas basadas en los hábitos registrados. | Las recomendaciones deben ajustarse a los datos de consumo del usuario y ser presentadas en menos de 2 segundos. | 90% - 100% |
| 9 | Prueba de Actualización de Recomendaciones | Validar que las recomendaciones cambian automáticamente en función de los nuevos datos. | Al ingresar nuevos datos, el sistema debe actualizar las recomendaciones en tiempo real. | 90% - 100% |
| 10 | Prueba de Alertas de Exceso de Nutrientes | Validar que el sistema genera alertas cuando el usuario excede los niveles recomendados. | El sistema debe generar una notificación automática cuando el consumo excede los límites establecidos. | 90% - 100% |
| 11 | Prueba de Configuración de Alertas | Verificar que el usuario pueda personalizar las alertas en horarios específicos. | El sistema debe permitir configurar las alertas para mostrarse en el momento seleccionado por el usuario. | 90% - 100% |
| 12 | Prueba de Creación y Edición de Perfiles | Comprobar que el sistema permite crear, editar y eliminar perfiles personalizados. | El usuario debe poder crear perfiles y estos deben guardarse correctamente. | 90% - 100% |
| 13 | Prueba de Restricciones Alimentarias | Validar que el sistema genera alertas basadas en las restricciones seleccionadas. | El sistema debe generar alertas o recomendaciones basadas en las restricciones seleccionadas. | 90% - 100% |
| 14 | Pruebas de Usabilidad (Beta Testing) | Validar que la aplicación es fácil de usar y las funcionalidades son intuitivas. | Los usuarios deben poder realizar tareas básicas sin asistencia, proporcionando feedback positivo sobre la experiencia. | 90% - 100% |
| 15 | Pruebas de Rendimiento | Comprobar que la aplicación tiene tiempos de respuesta rápidos, incluso bajo carga. | La aplicación debe mantener tiempos de respuesta inferiores a 3 segundos, incluso con múltiples usuarios concurrentes. | 90% - 100% |
| 16 | Pruebas de Integridad de Datos | Validar que no haya pérdida o corrupción de datos durante el uso de la aplicación. | Los datos ingresados deben guardarse correctamente y ser recuperados sin errores. | 90% - 100% |

### Control del alcance

El control del alcance asegura que se gestionen adecuadamente los cambios en el alcance del proyecto. Los cambios que se propongan deben pasar por un proceso formal de evaluación para determinar su impacto en el tiempo, costo y alcance.

Herramientas para el Control del Alcance del proyecto:

* Monday: Para gestionar tareas y hacer seguimiento de los cambios en los requisitos y el alcance.
* Reuniones de Revisión de Cambios: Se llevarán a cabo revisiones semanales para discutir cualquier cambio en el alcance y validar su aprobación.

Proceso de Control de Cambios:

* Cualquier solicitud de cambio en el alcance debe ser documentada formalmente según el formato 1 – REGISTRO DE CAMBIOS en carpeta de evidencia del proyecto.
* El equipo del proyecto evaluará el impacto del cambio.
* Los cambios aprobados se integrarán en el plan de trabajo y se comunicarán a los stakeholders.

### Cierre del Alcance

El cierre del alcance implica la verificación de que todo el trabajo requerido ha sido completado y aprobado. El proyecto no será considerado como terminado hasta que los stakeholders confirmen que los entregables cumplen con los requisitos definidos.

## Plan de gestión de los requerimientos

### Propósito del Plan de Gestión de Requerimientos

El propósito del Plan de Gestión de Requerimientos es definir y documentar los requerimientos del sistema para garantizar que el desarrollo de FoodWise cumpla con las expectativas y necesidades de los usuarios y stakeholders, en especial con los objetivos definidos por el profesor guía. Este plan proporciona una base para la planificación, ejecución y control de los requisitos durante todo el ciclo de vida del proyecto.

Este documento abarca los requisitos funcionales y no funcionales de la aplicación, así como la identificación de dependencias y criterios de aceptación para asegurar que cada funcionalidad sea correctamente implementada y validada.

### Definición de los Requerimientos del Proyecto

Compuesta por los requerimientos funcionales que describen las funciones y características que el sistema debe proporcionar para satisfacer las necesidades del usuario. Son aquellos requisitos que definen “qué” debe hacer la aplicación.

Por otro lado los requerimientos no funcionales: Establecen las cualidades del sistema, como su rendimiento, seguridad, usabilidad y compatibilidad. Son requisitos que definen “cómo” debe comportarse la aplicación y garantizar una buena experiencia de usuario.

### Listado de Requerimientos Funcionales

Los requerimientos funcionales de FoodWise establecen las capacidades que la aplicación ofrecerá a los usuarios para gestionar sus datos nutricionales y recibir recomendaciones personalizadas. Cada requisito busca facilitar el monitoreo de hábitos alimenticios y proporcionar alertas que promuevan un estilo de vida saludable.

|  |  |  |
| --- | --- | --- |
| **N°** | **ID** | **Requerimiento** |
| 1 | RF-1 | Ingreso manual de datos nutricionales de alimentos |
| 2 | RF-2 | Clasificación de alimentos por categorías |
| 3 | RF-3 | Creación de perfiles personalizados con metas nutricionales |
| 4 | RF-4 | Generación de recomendaciones personalizadas basadas en patrones de consumo |
| 5 | RF-5 | Proyecciones de bienestar y salud a largo plazo |
| 6 | RF-6 | Soporte para restricciones alimentarias (diabetes, celíacos, etc.) |
| 7 | RF-7 | Generación de alertas dinámicas para prevenir el consumo excesivo de nutrientes |
| 8 | RF-8 | Diseño de interfaz intuitiva y fácil de usar |
| 9 | RF-9 | Compatibilidad con gestos táctiles para navegación |
| 10 | RF-10 | Soporte para diferentes resoluciones de pantalla |
| 11 | RF-11 | Registro de usuarios con correo electrónico y contraseña |
| 12 | RF-12 | Recuperación de contraseña mediante correo electrónico |
| 13 | RF-13 | Generación de alertas sobre exceso de consumo de nutrientes (sodio, grasas, azúcares) |
| 14 | RF-14 | Notificaciones personalizadas para recordar la ingesta de alimentos o nutrientes específicos |
| 15 | RF-15 | Implementación de un modelo de árbol de decisiones para generar recomendaciones personalizadas |
| 16 | RF-16 | El modelo debe analizar patrones de consumo y hábitos del usuario para sugerir cambios alimenticios |

### Listado de Requerimientos No Funcionales

Los requerimientos no funcionales de FoodWise definen las características de rendimiento, seguridad y compatibilidad que deben cumplirse para garantizar una experiencia de usuario óptima y un funcionamiento seguro y fluido de la aplicación en múltiples plataformas.

|  |  |  |
| --- | --- | --- |
| N° | **ID** | **Requerimiento** |
| 1 | RNF 1 | La aplicación debe ser multiplataforma (iOS y Android) |
| 2 | RNF 2 | Escalabilidad para soportar el crecimiento de usuarios |
| 3 | RNF 3 | Seguridad en el manejo de datos personales y nutricionales |
| 4 | RNF 4 | Actualizaciones frecuentes sin interrupciones del servicio |
| 5 | RNF 5 | Optimización de rendimiento (carga rápida y fluidez en el uso) |
| 6 | RNF 6 | Compatibilidad con normativas alimentarias y de privacidad |
| 7 | RNF 7 | La interfaz debe ser responsiva y adaptarse a distintos tamaños de dispositivos |
| 8 | RNF 8 | Cifrado de contraseñas y protección de datos del usuario durante el registro |
| 9 | RNF 9 | Las alertas deben configurarse para mostrarse en horarios específicos elegidos por el usuario |
| 10 | RNF 10 | Las alertas deben ser precisas y no interrumpir el funcionamiento de otras funciones de la aplicación |
| 11 | RNF 11 | El modelo debe ser optimizado para que las decisiones se tomen en tiempo real sin afectar el rendimiento |

### Análisis de dependencias

Las dependencias son relaciones entre diferentes requisitos o fases del proyecto que indican que uno depende del cumplimiento de otro para poder ejecutarse correctamente:

**Registro de Usuarios y Seguridad:** La implementación de un sistema seguro de registro de usuarios (como el cifrado de contraseñas) depende de la funcionalidad de seguridad en el manejo de datos personales y nutricionales.

Diagrama

Descripción generada automáticamente

Ilustración 2: Análisis de dependencias Registro de Usuarios y Seguridad

**Usuario creado con restricciones alimentarias:** Esta es la etapa inicial donde el sistema verifica si el usuario ya está registrado y cuenta con restricciones alimentarias y preferencias previamente definidas (como ser celíaco, vegetariano, etc.). Esta información inicial permite que el sistema adapte la experiencia del usuario desde el principio.

Dependencias:

Ingreso de datos nutricionales: Depende de la existencia de un usuario registrado que pueda ingresar sus datos de consumo.

Registro de restricciones alimentarias: Esta información debe estar disponible desde el inicio para personalizar la experiencia de usuario.

**Ingreso de Datos Nutricionales:** El usuario ingresa información sobre su consumo diario de alimentos y nutrientes. estos datos son fundamentales para que el sistema pueda analizar patrones de consumo y generar recomendaciones basadas en los hábitos alimenticios específicos del usuario.

Dependencias:

Usuario creado: El usuario debe estar registrado y autenticado para ingresar datos.

Almacenamiento de datos: Los datos ingresados deben almacenarse y organizarse adecuadamente para ser utilizados en el análisis de patrones.

**Almacenamiento y Organización de Datos Nutricionales:** Esta etapa se encarga de organizar y almacenar de forma segura los datos nutricionales ingresados. Es un paso crucial para que el sistema pueda acceder a esta información en la fase de análisis.

Dependencias:

Ingreso de datos nutricionales: Depende de que el usuario ingrese su información de consumo para proceder con el almacenamiento.

Análisis de patrones de consumo: El análisis posterior depende de los datos correctamente organizados para identificar patrones.

**Registro de restricciones alimentarias y preferencias del usuario:** En esta etapa, se almacenan las restricciones y preferencias alimentarias del usuario (como celíaco, vegetariano, etc.), que se tomarán en cuenta al analizar los patrones de consumo y generar recomendaciones.

Dependencias:

Usuario creado con restricciones alimentarias: esta información debe estar registrada y almacenada desde el inicio para personalizar el análisis y las recomendaciones.

Análisis de patrones de consumo: este análisis debe considerar las restricciones para evitar recomendaciones de alimentos no adecuados.

**Análisis de patrones de consumo:** El sistema analiza los datos de consumo nutricional del usuario para identificar patrones o tendencias en su dieta. Este análisis es fundamental para generar recomendaciones basadas en el historial de consumo.

Dependencias:

Almacenamiento de datos nutricionales: Depende de que los datos estén correctamente organizados y accesibles.

Restricciones alimentarias: El análisis debe ajustarse para tener en cuenta las restricciones o preferencias del usuario, filtrando alimentos no adecuados según su perfil.

**Aplicación de modelo de árbol de decisiones con condiciones alimentarias:** Utilizando un modelo de árbol de decisiones, el sistema evalúa los patrones de consumo en combinación con las restricciones alimentarias para determinar recomendaciones específicas.

Dependencias:

Análisis de patrones de consumo: El árbol de decisiones necesita la salida del análisis de patrones para generar recomendaciones informadas.

Restricciones alimentarias: La lógica del modelo considera las restricciones alimentarias del usuario para personalizar aún más las recomendaciones.

**Generación de recomendaciones personalizadas:** Con base en los resultados del modelo de árbol de decisiones y el análisis de patrones, el sistema genera recomendaciones personalizadas que se ajustan a las condiciones del usuario. Estas recomendaciones pueden incluir ajustes en el consumo de nutrientes, recordatorios o sugerencias de alimentos alternativos.

Dependencias:

Aplicación del modelo de árbol de decisiones: las recomendaciones dependen de la salida del modelo de árbol de decisiones, que analiza los patrones de consumo y las restricciones alimentarias.

Restricciones y preferencias del usuario: las recomendaciones finales deben considerar estas restricciones para evitar sugerencias que no sean adecuadas para el usuario. alertas de consumo y configuración de horarios: las alertas sobre consumo excesivo dependen de la configuración de horarios específicos por parte del usuario, ya que estos definen cuándo se mostrarán las notificaciones.

Interfaz intuitiva y compatibilidad multiplataforma: la interfaz debe adaptarse a distintas resoluciones y plataformas (IOS y Android) antes de que se inicien las pruebas de usabilidad y optimización del rendimiento.

Diagrama

Descripción generada automáticamente

Ilustración 3: Análisis de dependencias para llegar a generación de recomendación

### Criterios de aceptación

Los criterios de aceptación definen las condiciones que cada funcionalidad debe cumplir para ser considerada completa y aprobada. Estos criterios serán utilizados en la fase de pruebas para validar el funcionamiento correcto de cada requisito.

|  |  |
| --- | --- |
| **Funcionalidad** | **Criterios de Aceptación** |
| USUARIO CREADO CON RESTRICCIONES ALIMENTARIAS | El sistema debe permitir la creación de un usuario con opciones para especificar restricciones alimentarias (celíaco, vegetariano, etc.) |
| La cuenta del usuario debe guardar de manera segura las restricciones y preferencias especificadas |
| Al acceder a la cuenta, el usuario debe ver un perfil con sus restricciones registradas |
| Las restricciones se aplican a todas las recomendaciones y alertas generadas |
| INGRESO DE DATOS NUTRICIONALES | El usuario debe poder ingresar datos nutricionales de forma manual para cada alimento consumido |
| Los datos deben incluir campos como calorías, proteínas, grasas, y azúcares |
| Confirmación de guardado sin errores de cada entrada nutricional |
| El usuario puede editar o eliminar entradas anteriores de datos nutricionales |
| ALMACENAMIENTO Y ORGANIZACIÓN DE DATOS NUTRICIONALES | Todos los datos nutricionales deben ser almacenados de forma organizada y segura |
| El sistema debe permitir el acceso a los datos para análisis futuros |
| Los datos deben estar organizados para facilitar la identificación de patrones |
| La organización debe permitir un acceso eficiente para el modelo de decisiones y análisis de patrones |
| REGISTRO DE RESTRICCIONES ALIMENTARIAS Y PREFERENCIAS DEL USUARIO | El sistema debe ofrecer opciones para seleccionar restricciones alimentarias durante el registro |
| Las restricciones deben ser almacenadas y reflejadas en todas las recomendaciones y alertas |
| Cualquier cambio en las restricciones debe aplicarse inmediatamente a las recomendaciones |
| El sistema filtra los alimentos no adecuados según el perfil del usuario |
| ANÁLISIS DE PATRONES DE CONSUMO | El sistema debe analizar los datos ingresados para identificar patrones de consumo |
| El análisis debe considerar las restricciones alimentarias y preferencias del usuario |
| Los patrones de consumo identificados deben ajustar las recomendaciones de forma automática |
| El análisis debe ejecutarse de manera eficiente sin afectar el rendimiento |
| APLICACIÓN DE MODELO DE ÁRBOL DE DECISIONES CON CONDICIONES ALIMENTARIAS | El modelo de árbol de decisiones debe utilizar patrones de consumo y restricciones alimentarias para evaluar el comportamiento nutricional |
| Las decisiones del modelo deben considerar las restricciones (por ejemplo, evitar gluten para celíacos) |
| Las recomendaciones generadas deben ser específicas y relevantes para el perfil del usuario |
| El modelo debe ejecutar recomendaciones en tiempo real sin afectar el rendimiento |
| GENERACIÓN DE RECOMENDACIONES PERSONALIZADAS | Las recomendaciones deben generarse considerando patrones de consumo y restricciones |
| Las recomendaciones deben ser claras y específicas, con sugerencias de ajustes en nutrientes |
| Se deben ofrecer alternativas para alimentos restringidos |
| El usuario debe recibir alertas personalizadas según las recomendaciones establecidas |

## Plan de gestión del cronograma

### Propósito de planificación del cronograma

Es establecer una estructura temporal clara y detallada que permita gestionar y controlar cada fase de desarrollo de manera eficiente. Dado el enfoque Waterfall, este cronograma asegura que cada fase se complete antes de pasar a la siguiente, garantizando una secuencia lógica y una documentación sólida en cada etapa.

Debemos definir claramente cada actividad del proyecto, desde la toma de requisitos hasta la implementación final y el mantenimiento, alineando el cronograma con los objetivos del sistema de recomendaciones personalizadas y restricciones alimentarias.

Debemos optimizar los recursos de manera eficiente en cada fase del proyecto, para asegurar que los desarrolladores, diseñadores y analistas estén disponibles en el momento adecuado, evitando cuellos de botella y garantizando el avance fluido de FoodWise.

Por otro lado, debemos crear una línea base temporal como punto de referencia para medir el desempeño de cada fase del proyecto. Esta línea base servirá para detectar desviaciones y asegurar que el proyecto permanezca dentro de los plazos establecidos.

Agregando, se debe asegurar que cada fase esté documentada y validada antes de avanzar, dado que Waterfall no permite modificaciones posteriores en fases anteriores. Esto es crucial para FoodWise, donde la precisión en la recopilación de requisitos y el diseño inicial afecta la funcionalidad final del sistema de recomendaciones.

Se debe priorizar desarrollar un proceso de seguimiento y control para identificar retrasos potenciales y gestionar cambios en el cronograma de forma organizada, manteniendo la comunicación constante con el profesor guía y los stakeholders del proyecto.

Finalmente, cumplir con la fecha de finalización establecida (30 de noviembre Aprox.), asegurando que cada fase avance según lo previsto, minimizando riesgos y maximizando el control sobre el tiempo de entrega de FoodWise.

### Definición y secuenciación de actividades

A continuación, se indica desglose de actividades por cada etapa acorde a la metodología Waterfall:

|  |  |  |
| --- | --- | --- |
| **Fase 1: Toma de Requerimientos y Documentación** | | |
| **Actividad** | **Descripción** | **Entregables** |
| Reunión inicial con stakeholders | Definir los objetivos y alcances del proyecto junto con los stakeholders y el equipo de proyecto. | Documento de objetivos y alcance inicial |
| Identificación de usuarios y roles | Determinar los perfiles de usuario y sus roles específicos dentro de la aplicación (por ejemplo, usuario general, con restricciones). | Documento de perfiles y roles de usuario |
| Documentación de requerimientos funcionales y no funcionales | Recopilar y documentar los requerimientos necesarios para el proyecto, tanto funcionales como no funcionales, asegurando claridad en cada especificación. | Documento de requerimientos |
| Validación y aprobación de requerimientos | Revisar el documento de requerimientos y validarlo con los stakeholders. | Documento de requerimientos aprobado |
| **Fase 2: Diseño del Sistema** | | |
| **Actividad** | **Descripción** | **Entregables** |
| Diseño de arquitectura del sistema | Definir la arquitectura general que incluya la base de datos, backend y frontend de la aplicación. | Documento de arquitectura del sistema |
| Diseño de interfaz de usuario y prototipado | Crear prototipos visuales de la interfaz utilizando herramientas como Figma, validándolos con los stakeholders. | Prototipo UI/UX aprobado |
| Diseño de la base de datos | Crear el modelo de la base de datos, definiendo tablas, relaciones y estructura para almacenar los datos nutricionales y de usuarios. | Diagrama de base de datos |
| Revisión y aprobación del diseño | Revisar y validar el diseño de arquitectura, base de datos e interfaz con el equipo y los stakeholders. | Diseño aprobado |
| **Fase 3: Implementación** | | |
| **Actividad** | **Descripción** | **Entregables** |
| Desarrollo del backend | Programar la lógica de negocio y configurar la conexión con la base de datos, así como las APIs necesarias para funcionalidades clave. | Backend funcional |
| Desarrollo del frontend | Crear la interfaz de usuario en React Native, incluyendo pantallas de usuario, navegación e interacción. | Frontend funcional |
| Integración del backend y frontend | Conectar el frontend y backend para que trabajen juntos y permitan la funcionalidad completa de la aplicación. | Aplicación integrada |
| Revisión de código y ajuste de funcionalidades | Realizar una revisión de código para asegurar la calidad, eficiencia y corrección de errores en la implementación. | Revisión de calidad de código |
| **Fase 4: Pruebas** | | |
| **Actividad** | **Descripción** | **Entregables** |
| Pruebas unitarias | Probar cada módulo del sistema de forma aislada para asegurar que cumplan con sus requerimientos. | Informe de pruebas unitarias |
| Pruebas de integración | Verificar que los módulos del sistema funcionan correctamente juntos, probando el flujo de datos y funcionalidad entre componentes. | Informe de pruebas de integración |
| Pruebas de aceptación | Realizar pruebas en conjunto con los stakeholders para verificar que la aplicación cumple con los requerimientos y expectativas iniciales. | Informe de pruebas de aceptación |
| **Fase 5: Implementación Final** | | |
| **Actividad** | **Descripción** | **Entregables** |
| Lanzamiento de la aplicación | Desplegar la aplicación en el entorno de producción, asegurando que esté disponible para los usuarios. | Aplicación en producción |
| Capacitación de usuarios (si aplica) | Proporcionar formación a los usuarios para asegurar que puedan utilizar la aplicación de manera efectiva. | Material de capacitación |
| Revisión final de funcionamiento | Verificar que todos los componentes y funcionalidades de la aplicación funcionen sin problemas en el entorno de producción. | Informe de revisión final |
| **Fase 6: Mantenimiento** | | |
| **Actividad** | **Descripción** | **Entregables** |
| Soporte y resolución de problemas | Responder a problemas y errores reportados por los usuarios, solucionándolos de manera oportuna. | Registro de incidencias y soluciones |
| Actualizaciones y mejoras | Implementar mejoras continuas para mantener la aplicación actualizada y funcional en base al feedback y necesidades emergentes. | Historial de actualizaciones y mejoras |

### Estimación de duración de las actividades

Esta se basa en el tiempo que se considera necesario para completar cada tarea según su complejidad, recursos y dependencias. Para esta estimación, utilizaremos un enfoque simplificado, pero preciso, considerando la disponibilidad del equipo y los métodos de estimación sugeridos por el PMBOK.

Los métodos de estimación son:

Estimación analógica: Basada en actividades similares realizadas en proyectos previos.

Juicio experto: Basado en el conocimiento y experiencia de los integrantes del grupo.

|  |  |  |
| --- | --- | --- |
| **Fase 1: Toma de Requerimientos y Documentación** | | |
| **Actividad** | **Duración Estimada** | **Método de Estimación** |
| Reunión inicial con stakeholders | 1 día | Estimación por Analogía |
| Identificación de usuarios y roles | 2 días | Estimación por Analogía |
| Documentación de requerimientos | 4 días | Estimación de Juicio de Expertos |
| Validación y aprobación de requerimientos | 2 días | Estimación por Analogía |
| **Fase 2: Diseño del Sistema** | | |
| **Actividad** | **Duración Estimada** | **Método de Estimación** |
| Diseño de arquitectura del sistema | 3 días | Estimación de Juicio de Expertos |
| Diseño de interfaz de usuario y prototipado | 5 días | Estimación por analogía |
| Diseño de la base de datos | 3 días | Estimación de Juicio de Expertos |
| Revisión y aprobación del diseño | 2 días | Estimación por Analogía |
| **Fase 3: Implementación** | | |
| **Actividad** | **Duración Estimada** | **Método de Estimación** |
| Desarrollo del backend | 10 días | Estimación por analogía |
| Desarrollo del frontend | 10 días | Estimación por analogía |
| Integración del backend y frontend | 5 días | Estimación de Juicio de Expertos |
| Revisión de código y ajuste de funcionalidades | 3 días | Estimación de Juicio de Expertos |
| **Fase 4: Pruebas** | | |
| **Actividad** | **Duración Estimada** | **Método de Estimación** |
| Pruebas unitarias | 5 días | Estimación por analogía |
| Pruebas de integración | 4 días | Estimación de Juicio de Expertos |
| Pruebas de aceptación | 3 días | Estimación por Analogía |
| **Fase 5: Implementación Final** | | |
| **Actividad** | **Duración Estimada** | **Método de Estimación** |
| Lanzamiento de la aplicación | 2 días | Estimación por Analogía |
| Capacitación de usuarios (si aplica) | 3 días | Estimación de Juicio de Expertos |
| Revisión final de funcionamiento | 2 días | Estimación por Analogía |
| **Fase 6: Mantenimiento** | | |
| **Actividad** | **Duración Estimada** | **Método de Estimación** |
| Soporte y resolución de problemas | Continuo | N/A |
| Actualizaciones y mejoras | Continuo | N/A |

### Desarrollo del cronograma

Para el desarrollo del cronograma, se ha seguido un proceso de planificación, en la que cada fase en actividades específicas, alineadas con la Estructura de Desglose del Trabajo (EDT) para asegurar que todas las tareas necesarias estén claramente identificadas. La secuenciación de actividades está organizada en un orden lógico, respetando las dependencias entre ellas para garantizar que cada fase sea completada antes de avanzar.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Fase** | **Actividad** | **Duración** | **Fecha de Inicio** | **Fecha de Término** |
| **Fase 1: Toma de Requerimientos y Documentación** | Reunión inicial con stakeholders | 1 día | 01-sept-24 | 01-sept-24 |
| Identificación de usuarios y roles | 2 días | 02-sept-24 | 03-sept-24 |
| Documentación de requerimientos | 4 días | 04-sept-24 | 09-sept-24 |
| Validación y aprobación de requerimientos | 2 días | 10-sept-24 | 11-sept-24 |
| **Fase 2: Diseño del Sistema** | Diseño de arquitectura del sistema | 3 días | 12-sept-24 | 16-sept-24 |
| Diseño de interfaz de usuario y prototipado | 5 días | 17-sept-24 | 23-sept-24 |
| Diseño de la base de datos | 3 días | 24-sept-24 | 26-sept-24 |
| Revisión y aprobación del diseño | 2 días | 27-sept-24 | 30-sept-24 |
| **Fase 3: Implementación** | Desarrollo del backend | 10 días | 01-oct-24 | 14-oct-24 |
| Desarrollo del frontend | 10 días | 01-oct-24 | 14-oct-24 |
| Integración del backend y frontend | 5 días | 15-oct-24 | 21-oct-24 |
| Revisión de código y ajuste de funcionalidades | 3 días | 22-oct-24 | 24-oct-24 |
| **Fase 4: Pruebas** | Pruebas unitarias | 5 días | 25-oct-24 | 31-oct-24 |
| Pruebas de integración | 4 días | 01-nov-24 | 06-nov-24 |
| Pruebas de aceptación | 3 días | 07-nov-24 | 11-nov-24 |
| **Fase 5: Implementación Final** | Lanzamiento de la aplicación | 2 días | 12-nov-24 | 13-nov-24 |
| Capacitación de usuarios (si aplica) | 3 días | 14-nov-24 | 16-nov-24 |
| Revisión final de funcionamiento | 2 días | 18-nov-24 | 19-nov-24 |
| **Fase 6: Mantenimiento** | Soporte y resolución de problemas | 5 días | 20-nov-24 | 26-nov-24 |
| Actualizaciones y mejoras finales | 4 días | 27-nov-24 | 30-nov-24 |

### Control del cronograma

A través de un seguimiento continuo y del uso de la plataforma de gestión de proyectos Monday, se va a monitorear el avance de cada fase, identificar problemas tempranamente y tomar decisiones informadas para mantener el proyecto en curso.

Dentro de los objetivos del control del cronograma se señalan:

* Asegurar que cada fase se complete dentro de los tiempos establecidos.
* Identificar y gestionar de manera eficiente cualquier desviación en el cronograma.
* Facilitar la comunicación y el flujo de información en tiempo real entre los miembros del equipo y los stakeholders.

Monday será la herramienta central para gestionar y monitorear el cronograma. En esta plataforma, se registrarán todas las actividades planificadas, asignaciones de tareas, dependencias y fechas clave, proporcionando un entorno colaborativo y visual que facilita el seguimiento diario del proyecto.

Dentro de las estrategias y procesos de control, se establece la línea base del cronograma que estará indicado en Monday, donde se indicará el inicio y termino de cada fase. Esta línea base servirá como punto de referencia para medir el desempeño real frente al plan.

Monday permite actualizar en tiempo real las actividades, indicando si están en progreso, completadas o retrasadas, y se pueden programar alertas automáticas.

El análisis de las desviaciones estará a cargo de Ayax Rebolledo, quién analizará la raíz o causa y evaluar el impacto ¿en el cronograma general.

Se generará un plan de recuperación, reasignando recursos o redistribución de actividades. Los cambios necesarios en el cronograma serán reflejados en Monday y comunicados a todo el equipo.

No se generarán informes de avances, ya que estos quedarán evidenciados en tiempo real en Monday.

### Responsabilidades clave y gestión de cambios en el plan de gestión del cronograma

El líder de proyecto Ayax Rebolledo será el responsable de monitorear el cronograma en Monday, y el análisis de desviaciones, para la toma de decisiones.

El equipo total y stakeholders actualizarán respecto al estado de las actividades y revisión periódica del progreso.

Los cambios en el cronograma serán acordados y establecidos entre el equipo desarrollador y líder del proyecto.

## Plan de gestión de Costos

### Propósito del plan de gestión de costos

El Plan de Gestión de Costos del proyecto FoodWise establece los procedimientos para planificar, estimar, gestionar y controlar los costos a lo largo del ciclo de vida del proyecto. Este documento garantiza que el proyecto se mantenga dentro del presupuesto aprobado y que se utilicen los recursos financieros de manera eficiente.

### Estimación de costos

Costos operativos principales:

* Publicidad: $500,000 anual, incrementa 3% anual.
* Servicios en la nube: $1,200,000 en el primer año, con un aumento del 3% anual.

Otros costos recurrentes: Basados en la naturaleza operativa mensual y anual, con ajustes proyectados para inflación y cambios del mercado.

### Presupuesto

El presupuesto total del proyecto se construye considerando los ingresos esperados y los costos operacionales:

|  |  |  |  |
| --- | --- | --- | --- |
| **Año** | **Ingresos ($)** | **Costos operacionales ($)** | **Resultado Neto ($)** |
| Año 1 | $ 703 | 1,700,000 | $ -997 |
| Año 2 | $ 2.460.360 | $ 1.751.000 | $ 709 |
| Año 3 | $ 3.602.670 | $ 1.803.530 | $ 1.799.140 |

### Control de Costos

El control se llevará a cabo mediante la comparación continua entre el presupuesto planificado y los costos reales, utilizando las siguientes herramientas:

* Flujos de caja proyectados para detectar desviaciones.
* Revisiones mensuales del presupuesto para ajustes.
* Alertas automáticas cuando los costos superen el 10% del presupuesto aprobado.

### Indicadores clave de desempeño (KPI)

Para asegurar que el proyecto FoodWise se mantenga dentro del presupuesto y maximice el uso eficiente de los recursos financieros, se han definido indicadores clave de desempeño (KPIs) específicos. Estos KPIs permiten evaluar el progreso financiero del proyecto de manera objetiva, identificando posibles desviaciones a tiempo y permitiendo tomar medidas correctivas de ser necesario. Los indicadores seleccionados proporcionan una visión detallada de la salud financiera del proyecto en relación con el presupuesto aprobado, ayudando al equipo a cumplir con los objetivos de costos y manteniendo la rentabilidad proyectada.

Los indicadores claves de desempeño identificados son:

* Variación de costo (CV): CV = Valor Ganado (EV) - Costo Real (AC).
* Índice de Desempeño de Costo (CPI): CPI = EV / AC. Un CPI < 1 indica sobrecostos.
* \*ROI ajustado al riesgo: Se monitorea anualmente para verificar la rentabilidad.

### Roles y responsabilidades

Equipo de Desarrollo (Ayax Rebolledo y Jordán Rosas)

Función: Proveer información sobre los recursos necesarios para completar las tareas planificadas en cada sprint.

Relación con costos: Identifican desviaciones de recursos (como servicios en la nube o necesidades técnicas) que puedan afectar los costos operativos o el cronograma. Reportan estos hallazgos al Product Owner para ajustes en el backlog.

Product Owner (PO):

Función: Priorización de funcionalidades con foco en maximizar el valor entregado dentro del presupuesto aprobado.

Relación con costos: Alinea las decisiones del backlog con el presupuesto disponible y ajusta las prioridades si se detectan desviaciones en los costos o necesidades adicionales de recursos.

### Procesos de aprobación de cambios

Los cambios en el presupuesto requieren:

* Solicitud formal mediante el comité de control de cambios.
* Evaluación del impacto sobre las líneas base del costo.
* Aprobación del sponsor antes de su implementación (en caso de conseguir sponsor en un futuro).

### Herramientas y reportes financieros

Para realizar el seguimiento cercano de los ingresos, costos, y cualquier desviación en relación con el presupuesto aprobado se usarán las herramientas más abajo detalladas. Estas herramientas y reportes ofrecen visibilidad en tiempo real sobre el estado financiero del proyecto, permitiendo identificar patrones de desempeño y tomar decisiones informadas que favorezcan el cumplimiento de los objetivos financieros.

* Herramienta de caja de flujo: monitorear de manera continua los ingresos y costos del proyecto. Con la caja de flujo, el equipo de gestión puede observar las entradas y salidas de dinero en el proyecto, asegurando que los recursos se mantengan alineados con el presupuesto y anticipando posibles necesidades de ajuste en el flujo de efectivo.
* Reportes trimestrales: Emitidos cada tres meses, los reportes financieros proporcionan un análisis exhaustivo de las métricas clave y cualquier desviación que haya ocurrido respecto al presupuesto. Estos reportes ofrecen a los stakeholders una visión detallada del desempeño financiero, permitiendo evaluar el progreso hacia las metas de rentabilidad y ajustando estrategias cuando sea necesario.

## Plan de gestión de Calidad

### Propósito del plan de calidad

El plan de calidad tiene como objetivo garantizar que todos los entregables del proyecto FoodWise cumplan con los estándares establecidos, siguiendo los principios de la metodología Waterfall. Esto incluye verificar que cada fase (requisitos, diseño, implementación, pruebas y despliegue) se complete con la calidad esperada antes de avanzar a la siguiente.

### Objetivos de la calidad

* Asegurar que el proyecto cumpla con los requisitos funcionales y no funcionales definidos.
* Minimizar los errores mediante una revisión exhaustiva en cada fase.
* Cumplir con los estándares internacionales de calidad en software, como ISO/IEC 25010.
* Garantizar la satisfacción de los interesados, ofreciendo una solución robusta y alineada con los objetivos iniciales.

### Criterios o estándares de calidad

Estos criterios se agrupan en categorías clave como funcionalidad, usabilidad, rendimiento y seguridad. Se derivan de los requisitos del proyecto y sirven como guías para el diseño, desarrollo, pruebas y validación del sistema:

* Funcionalidad, que está definida en los requerimientos funcionales y no funcionales especificados en el apartado “plan de requerimientos”.
* Usabilidad, la cual enfocamos en la experiencia del usuario (UX) y la interfaz gráfica (UI).
* Rendimiento, que está definido por la eficiencia del sistema.
* Seguridad, definida por el Firebase Authentication, ya que utiliza algoritmos seguros (como bcrypt) para almacenar las contraseñas de los usuarios. Además se agrega una validación de contraseña segura, la cual debe tener una longitud de 8 caracteres, donde deben ir incluidos un signo, un número y una mayúscula.

### Enfoque de la calidad

El enfoque está basado en las etapas del ciclo Waterfall:

#### Fase de requisitos

1. Verificar que los requisitos sean claros, completos y aprobados por los stakeholders.
2. Definir criterios de aceptación detallados para cada funcionalidad.

#### Fase de diseño

1. Revisar la arquitectura técnica y los prototipos.
2. Validar que el diseño cubra todos los requisitos documentados.

#### Fase de implementación

1. Aplicar estándares de codificación y realizar revisiones de código.
2. Garantizar la cobertura de pruebas unitarias (>80%).

#### Fase de pruebas

1. Realizar pruebas funcionales, de integración y de rendimiento.
2. Validar todos los casos de uso definidos.

#### Fase de despliegue

1. Revisar todos los entregables antes del despliegue.
2. Validación final con los stakeholders.

### Estándares y criterios de aceptación por fase

|  |  |  |
| --- | --- | --- |
| **Fase** | **Estándares** | **Criterios de Aceptación** |
| **Requisitos** | Requisitos claros, completos y sin ambigüedades. | Todos los requisitos aprobados por los stakeholders antes de pasar al diseño. |
| Documentación en un formato estándar, legible y comprensible. |  |
| **Diseño** | Diseños alineados con los requisitos funcionales. | Aprobación del diseño por el equipo técnico y los stakeholders. |
| Diagramas arquitectónicos claros (como diagramas de flujo, clases y componentes). | Documentación del diseño completa antes de avanzar. |
| Uso de estándares internacionales como ISO/IEC 25010. |  |
| **Implementación** | Código siguiendo buenas prácticas de programación (Clean Code). | Revisión de código cruzada (peer reviews). |
| Uso de herramientas para análisis estático de código (SonarQube, ESLint). | Cobertura de pruebas unitarias mayor al 80%. |
| Modularidad del código para facilitar pruebas y mantenimiento. |  |
| **Pruebas** | Documentación de todos los casos de prueba (funcionales, integración y rendimiento). | Todos los casos de prueba ejecutados sin defectos críticos. |
| Uso de herramientas automatizadas para pruebas funcionales y de rendimiento (Jest, Postman). | Todos los defectos documentados deben ser corregidos antes del despliegue. |
| Validación de las interfaces en dispositivos iOS y Android. |  |
| **Despliegue** | Pruebas finales realizadas en el entorno de producción. | Validación final por stakeholders antes del lanzamiento. |
| Garantizar el cumplimiento de requisitos no funcionales (rendimiento, seguridad). | Documentación completa del proceso de entrega. |

### Responsabilidades

|  |  |
| --- | --- |
| **Rol** | **Responsabilidad** |
| **Jefe de Proyecto** | Supervisar que cada fase cumpla con los estándares de calidad. |
| **Equipo de Desarrollo** | Aplicar buenas prácticas de codificación y realizar pruebas unitarias. |
| **Tester** | Diseñar y ejecutar pruebas exhaustivas. |
| **Stakeholders** | Revisar y aprobar los entregables en cada fase. |

## Plan de gestión de Riesgos

### Propósito del plan de riesgos

El propósito del Plan de Riesgos del proyecto FoodWise es identificar, evaluar y gestionar proactivamente los riesgos que puedan afectar el éxito del proyecto. Este plan busca minimizar el impacto negativo de los riesgos y maximizar las oportunidades, garantizando que el proyecto se mantenga alineado con sus objetivos en cuanto a desarrollo, cronograma y calidad del producto final. La gestión de riesgos también pretende proteger la seguridad y privacidad de los datos de los usuarios, asegurar el cumplimiento de los objetivos de salud y bienestar establecidos, y fomentar la adaptación y satisfacción de los usuarios con la aplicación.

### Definición del plan de riesgos

El Plan de Riesgos es un documento formal que detalla el proceso de identificación, evaluación, respuesta y monitoreo de los riesgos asociados con el proyecto. Este plan incluye una lista exhaustiva de posibles riesgos, su categorización según probabilidad e impacto, y las estrategias de respuesta definidas para cada uno. Además, establece las responsabilidades del equipo en la gestión de riesgos y las metodologías a emplear para el seguimiento y monitoreo de estos riesgos a lo largo de la ejecución del proyecto. El objetivo es contar con un enfoque estructurado que permita anticiparse a los problemas potenciales y tomar decisiones informadas para mitigar sus efectos.

### Desgloses de riesgos por área

#### Riesgos técnicos

* Fallas de integración o problemas al integrar la app móvil con el backend y la base de datos.
* Problemas causados por una mala toma de requerimientos, que afecten en etapas avanzadas del proyecto el alcance del mismo.
* Posibilidad de que la infraestructura no soporte el crecimiento de usuarios, afectando el rendimiento.
* Riesgo de incompatibilidad de React Native con ciertas versiones de iOS o Android.
* Riesgo de que el motor de recomendaciones y predicciones no cumpla con los objetivos de personalización.
* Riesgo de errores o fallas en funcionalidades clave.
* Riesgo de que las recomendaciones y alertas no sean precisas o útiles, afectando la percepción de la app.
* Datos de alimentos o recomendaciones podrían ser inexactos, afectando la confiabilidad.
* Interrupciones o riesgo de caídas o problemas de conectividad.
* Posibilidad de que el servidor o base de datos no escalen adecuadamente ante un aumento de usuarios.
* Complejidad de integración entre las diversas partes de la arquitectura del sistema.

#### Riesgos Externos

* Variación de la UF o del dólar, afectando costo de servicios externos que pueda necesitar el proyecto.
* Exposición de datos debido a configuraciones inadecuadas o ataques externos.
* Riesgo de no cumplir con normativas locales o internacionales de seguridad alimentaria.
* Riesgo legal si los usuarios consideran que las recomendaciones impactan negativamente su salud.

#### Riesgos organizacionales

* Posibilidad de no contar con el personal o presupuesto necesario.
* Riesgo de dificultades en el mantenimiento a largo plazo, afectando la capacidad para mejorar el producto.
* Riesgo de no priorizar de forma adecuada, las diversas tareas asociadas al proyecto, afectando las líneas base del proyecto.

#### Riesgos de dirección de proyecto

* Riesgo de retraso y de no completar las fases del proyecto según el cronograma.
* Desviaciones en el alcance del proyecto, lo que afectaría el tiempo y los recursos.
* Problemas asociados a una mala comunicación entre el equipo, afectando y retrasando el proyecto.
* Un mal control del tiempo, costo y alcance, que afecten la calidad del proyecto.
* Estimación errónea respecto al tiempo dedicado a cada tarea del proyecto.

### Estructura de desglose de riesgos RBS

Diagrama

Descripción generada automáticamente

Ilustración 4: Estructura de desglose de riesgos RBS

### Matriz de riesgos

## Control integrado de cambios

## Control de costos y presupuestos

## Control de riesgos

## Control de recursos

## Control de Cronograma e Hitos

## Informe de estado de avance

## Estado Avance Costos y Presupuesto

## Estado Avance Recursos

## Estado Avance Cronograma e Hitos

## Acta de cierre del proyecto