**Acta de Constitución**

**del Proyecto**

***[NutriMove]***

***Fecha: [02/09/2024]***

**Tabla de contenido**

[**Propósito y Justificación del Proyecto 2**](#_heading=h.3znysh7)

[**Descripción del Proyecto 3**](#_heading=h.2et92p0)

[**Requerimientos de alto nivel 4**](#_heading=h.prjcib276rut)

[**Objetivos 6**](#_heading=h.4d34og8)

[**Riesgos iniciales de alto nivel 8**](#_heading=h.17dp8vu)

[**Cronograma de hitos principales 9**](#_heading=h.3rdcrjn)

[**Presupuesto estimado 10**](#_heading=h.26in1rg)

[**Lista de Interesados (stakeholders) 10**](#_heading=h.kqfwe0mlcbe6)

[**Asignación del Gerente de Proyecto y nivel de autoridad 10**](#_heading=h.1ksv4uv)

[**Personal y recursos preasignados 11**](#_heading=h.z337ya)

[**Aprobaciones 11**](#_heading=h.3j2qqm3)

**Información del Proyecto**

**Datos**

| Empresa / Organización | Estudiantes Duoc |
| --- | --- |
| Proyecto | NutriMove |
| Fecha de preparación | 02/09/2024 |
| Cliente | Público General |
| Patrocinador principal | \* |
| Gerente de Proyecto | Miguel Ravello |

# Propósito y Justificación del Proyecto

| El proyecto usará inteligencia artificial para generar rutinas de ejercicio físico y planes de alimentación personalizadas, diferenciándose de otras aplicaciones al adaptarse a las necesidades específicas de cada usuario y evitando la generalización. |
| --- |

# Descripción del Proyecto

| Desarrollo de una Aplicación Interactiva para Planes de Entrenamiento y Alimentación Personalizados  Se está desarrollando una aplicación interactiva que genera planes de entrenamiento físico personalizados y de libre acceso, con la opción de un plan alimenticio complementario para optimizar la evolución del usuario. El sistema se implementará utilizando una arquitectura de microservicios, empleando FastAPI para el back-end y Docker para contenerizar el proyecto.  Arquitectura del Sistema RAG   1. Microservicio de Generación de Plan de Entrenamiento: Crea planes personalizados basados en las preferencias y objetivos del usuario. 2. Microservicio de Generación de Plan Alimenticio: Genera planes alimenticios alineados con los entrenamientos y necesidades nutricionales del usuario. 3. Microservicio de Integración y Optimización: Asegura que los planes de entrenamiento y alimentación estén alineados para cumplir los objetivos del usuario. 4. Microservicio de Almacenamiento y Recuperación de Información: Gestiona el almacenamiento y la recuperación de los datos del usuario y sus progresos. 5. Sistema de Recuperación y Generación de Contenidos (RAG): Combina la recuperación de información con la generación de contenido para ofrecer respuestas personalizadas y contextuales.   Dockerización del Proyecto:   * Se dockerizarán los microservicios para asegurar un despliegue eficiente y escalable, con la posibilidad de utilizar Kubernetes para la orquestación de los contenedores. |
| --- |

# Requerimientos de alto nivel

**Requerimientos del proyecto**

| Requerimientos Funcionales  1. **Generación de Planes de Entrenamiento:**    * El sistema debe permitir a los usuarios crear planes de entrenamiento personalizados.    * Debe ofrecer opciones para seleccionar el tipo de entrenamiento (fuerza, resistencia, flexibilidad, etc.).    * Debe permitir la personalización del plan según el nivel de condición física y objetivos del usuario. 2. **Generación de Planes Alimenticios:**    * El sistema debe generar planes alimenticios que complementen los planes de entrenamiento.    * Debe tener en cuenta las preferencias alimentarias y restricciones dietéticas de los usuarios.    * El plan alimenticio debe estar alineado con los objetivos de entrenamiento (pérdida de peso, ganancia de músculo, etc.). 3. **Integración de Planes de Entrenamiento y Alimentación:**    * Los planes de entrenamiento y alimentación deben estar sincronizados para optimizar los resultados del usuario.    * El sistema debe proporcionar recomendaciones de ajustes en los planes en función del progreso del usuario. 4. **Gestión de Usuarios:**    * El sistema debe permitir a los usuarios crear y gestionar perfiles con sus datos personales, preferencias y objetivos.    * Debe almacenar el historial de entrenamiento y dieta para ofrecer un seguimiento del progreso. 5. **Sistema de Recuperación y Generación de Contenidos (RAG):**    * Debe integrar técnicas de recuperación de información para mejorar la personalización y precisión de los planes generados.    * Debe ser capaz de ajustar dinámicamente los planes basados en nuevos datos o cambios en las preferencias del usuario. 6. **Interfaz de Usuario:**    * La aplicación debe contar con una interfaz amigable que permita a los usuarios interactuar fácilmente con el sistema.    * Debe ofrecer una visualización clara de los planes de entrenamiento y alimentación, así como del progreso.  Requerimientos No Funcionales  1. **Escalabilidad:**    * La aplicación debe ser escalable para soportar un número creciente de usuarios y datos.. 2. **Desempeño:**    * El sistema debe responder rápidamente a las solicitudes de los usuarios.    * Los microservicios deben estar optimizados para un rendimiento eficiente. 3. **Seguridad:**    * El sistema debe garantizar la seguridad de los datos del usuario, incluyendo la protección de la información personal y nutricional.    * Debe implementar autenticación y autorización robustas para proteger el acceso a la información sensible. 4. **Portabilidad:**    * Los microservicios deben ser contenerizados usando Docker, permitiendo su despliegue en diferentes entornos de manera consistente.    * Se debe considerar el uso de una herramienta de orquestación de contenedores para gestionar los recursos de manera eficiente y facilitar la escalabilidad del sistema. 5. **Mantenibilidad:**    * El código del proyecto debe ser modular y fácil de mantener, permitiendo actualizaciones y mejoras sin afectar el sistema en su totalidad.    * La documentación del código y de la arquitectura debe ser clara y detallada para facilitar futuras modificaciones. 6. **Disponibilidad:**    * El sistema debe estar disponible 24/7 para los usuarios, con mecanismos para la recuperación ante fallos.    * Debe implementarse un monitoreo continuo para detectar y resolver problemas rápidamente. |
| --- |

# Objetivos

# 

| **Objetivo** | **Indicador de éxito** |
| --- | --- |
| **Alcance** | |
| Gestión de Parámetros de Entrada | La app logra captar correctamente los parámetros necesarios para generar la rutina personalizada del usuario ajustándose a estos después de haber finalizado las siguiente etapas: el levantamiento funcional, desarrollo, QA y finalmente la puesta en marcha |
| Configuración y Seguimiento de Rutinas | Genera correctamente el plan de entrenamiento personalizado que se ajusta a la necesidad del usuario y recopila el avance del usuario, una vez finalizadas las siguiente etapas: el levantamiento funcional, desarrollo, QA y finalmente la puesta en marcha. |
| Visualización de Estadísticas | Genera un dashboard que permite visualizar el avance del usuario en base al rendimiento de este en su plan de entrenamiento, una vez finalizadas las siguiente etapas: el levantamiento funcional, desarrollo, QA y finalmente la puesta en marcha |
| Gestión de Plan Alimentación | Genera correctamente el plan de alimentación acorde al entrenamiento y necesidades del usuario, después de haber terminado los siguientes hitos: Levantamiento funcional, desarrollo, QA y finalmente la puesta en marcha. |
| Plan de alimentación | Desarrolla un plan nutricional personalizado alineado con los objetivos del usuario, y lo adapta de manera continua basándose en el avance en su rutina de entrenamiento.. |
| Seguimiento y control | Se mantiene un control constante del estado del proyecto mediante informes de progreso, revisiones de calidad y actas de reunión, para asegurar que las metas se cumplen en tiempo y forma, incluyendo la captación de feedback de los usuarios. |
| Plan de despliegue | Define los pasos detallados para llevar a cabo la implementación de la app, asegurando una transición sin problemas desde el entorno de desarrollo al entorno de producción, incluyendo planes de contingencia y rollback. |
| Cierre de proyecto | Se completa el proyecto con la entrega final de todas las funcionalidades, la aceptación del cliente y la documentación de lecciones aprendidas para futuras mejoras. |
| Planificación | Se establecen las fases clave del proyecto (levantamiento de requerimientos, desarrollo, pruebas, implementación), el cronograma, recursos y riesgos a gestionar. |
| Desarrollo | Se construyen las funcionalidades de la app, desde la gestión de parámetros del usuario hasta la creación de las rutinas de entrenamiento y planes alimenticios, con código bien documentado y probado. |
| Pruebas | Se realizan pruebas unitarias, de integración y aceptación para garantizar que la app funcione correctamente y se ajuste a las necesidades del usuario, asegurando calidad en cada etapa antes de la puesta en marcha. |
| Implementación de base de datos | Se diseña y despliega una base de datos eficiente que gestiona los parámetros de los usuarios, sus rutinas de entrenamiento y planes alimenticios, asegurando la integridad y disponibilidad de la información. |
| Origen ágil | Se adopta una metodología ágil (Scrum) para permitir iteraciones rápidas, mejoras continuas y ajustes en el proyecto según las necesidades del usuario y el feedback recibido durante el desarrollo. |
| **Cronograma (Tiempo)** | |
| Gestión de Parámetros de Entrada | 17 días |
| Configuración y Seguimiento de Rutinas | 20 días |
| Visualización de Estadísticas | 17 días |
| Gestión de Alimentación | 24 días |
| Plan de alimentación | 23 días |
| Seguimiento y control | 89 días |
| Plan de despliegue | 14 días |
| Cierre de proyecto | 6 días |
| Planificación | 7 días |
| Desarrollo | 78 días |
| Pruebas | 14 días |
| Implementación de base de datos | 20 días |
| Origen ágil | 7 días |
| **Costo** | |
| Gestión de Parámetros de Entrada | CLP $2,720,000 |
| Configuración y Seguimiento de Rutinas | CLP $3,200,000 |
| Visualización de Estadísticas | CLP $2,720,000 |
| Gestión de Alimentación | CLP $3,840,000 |
| Plan de alimentación | CLP $3,680,000 |
| Seguimiento y control | CLP $14,240,000 |
| Plan de despliegue | CLP $2,240,000 |
| Cierre de proyecto | CLP $960,000 |
| Planificación | CLP $1,120,000 |
| Desarrollo | CLP $12,480,000 |
| Pruebas | CLP $2,240,000 |
| Implementación de base de datos | CLP $3,200,000 |
| Origen ágil | CLP $1,120,000 |
| **Calidad** | |
| Gestión de Parámetros de Entrada | La generación de la rutina toma en cuenta los parámetros ingresados por el usuario y los aplica para adaptarse lo mejor posible a las necesidades del usuario. |
| Configuración y Seguimiento de Rutinas | El plan se ajusta a lo solicitado por el usuario y recopila el progreso de este para proporcionar el seguimiento del progreso de este siguiendo la rutina proporcionada. |
| Visualización de Estadísticas | Se logra visualizar correctamente el progreso del usuario según el rendimiento de este siguiendo el plan de entrenamiento. |
| Gestión de Alimentación | El plan de alimentación se ajusta al plan de entrenamiento y a las condiciones medicas y/o físicas del usuario. |
| Plan de alimentación | Se logra almacenar correctamente el plan de alimentación aparte se adapta a las necesidades del usuario. |
| Seguimiento y control | Se logra mantener un seguimiento y control de del proyecto logrando adaptarse a los cambios y solucionando los errores que se presentaron aparte de mantener documentado el proceso correctamente. |
| Plan de despliegue | Se logra generar un plan de despliegue exitoso y se logra documentar correctamente |
| Cierre de proyecto | Se logra un cierre de proyecto exitoso y se documenta correctamente. |
| Planificación | Se documenta correctamente la planificación. |
| Desarrollo | Se logra documentar y desarrollar el codigo del proyecto. |
| Pruebas | Se realizan correctamente las pruebas corroborando que la app funcione correctamente. |
| Implementación de base de datos | Se logra crear correctamente la base de datos que cumple con las necesidades del proyecto. |
| Origen ágil | Se logra obtener las historias de usuario lo que ayuda al desarrollo. |

# Riesgos iniciales de alto nivel

| **Levantamiento funcional** | 1. Malentendidos en los requisitos. 2. Cambios frecuentes. 3. Problemas de usabilidad y experiencia de usuario. |
| --- | --- |
| **Desarrollo** | 1. Errores en la integración de servicios externos. 2. Errores en la interfaz gráfica. 3. Errores en lógica, algoritmos y cálculos. |
| **Testeo** | 1. Falta de planificación. 2. Dejar las pruebas para el final. 3. No documentar las pruebas. |
| **Levantamiento de requerimientos** | 1. Mala toma de requerimientos. 2. Identificación errónea de los requerimientos. 3. Exposición a los cambios de los requerimientos /Alcance. |

# Cronograma de hitos principales

| **Hito** | **Fecha tope** |
| --- | --- |
| Gestión de Parámetros de Entrada | 15-09-2024 |
| Configuración y Seguimiento de Rutinas | 10-15-2024 |
| Visualización de Estadísticas | 01-11-2024 |
| Gestión de Alimentación | 25-11-2024 |
| Plan de alimentación | 25-09-2024 |
| Seguimiento y control | 30-11-2024 |
| Plan de despliegue | 25-11-2024 |
| Cierre de proyecto | 01-12-2024 |
| Planificación | 08-09-2024 |
| Desarrollo | 25-11-2024 |
| Pruebas | 25-11-2024 |
| Implementación de base de datos | 28-09-2024 |
| Origen ágil | 08-09-2024 |

# Presupuesto estimado

| El presupuesto es en primera instancia será de $10.000.000 |
| --- |

# Lista de Interesados (stakeholders)

| **Nombre** | **Cargo** | **Departamento / División** |
| --- | --- | --- |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

# Asignación del Gerente de Proyecto y nivel de autoridad

**Gerente de Proyecto**

| **Nombre** | **Cargo** | **Departamento / División** |
| --- | --- | --- |
| **Miguel Ravello** | Product manager (PM) | Tecnologías de la Información (TI) |

**Niveles de autoridad**

| **Área de autoridad** | **Descripción del nivel de autoridad** |
| --- | --- |
| Gestión de presupuesto y de sus variaciones | Alto: Maneja el presupuesto y toma de decisiones  en vista de los gastos. Pudiendo limitar o solicitar más  . |
| Decisiones técnicas | Alto : Encargado en la toma decisiones enfocadas en las áreas técnicas del proyecto. |
| Resolución de conflictos | Alto: Es responsable de desarrollar e implementar planes y procedimientos, así como políticas de personal. |
| Ruta de escalamiento y limitaciones de autoridad | Alto: Encargado de establecer límites en el desarrollo. |

# Personal y recursos preasignados

| **Recurso** | **Departamento / División** |
| --- | --- |
| Miguel Ravello | TI |
| Felipe Prieto | TI |
| Alejando Vasquez | TI |
| Patricio Catejo | TI |

# Aprobaciones

| **Patrocinador** | **Fecha** | **Firma** |
| --- | --- | --- |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |