**Acta de Constitución**

**del Proyecto**

***[Nombre del Proyecto]***

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

**Tabla de contenido**

Información del Proyecto 3

Datos 3

Patrocinador / Patrocinadores 3

Propósito y Justificación del Proyecto 3

Descripción del Proyecto y Entregables 3

Requerimientos de alto nivel

Requerimientos del proyecto 4

Objetivos 4

Premisas y Restricciones 5

Riesgos iniciales de alto nivel 5

Cronograma de hitos principales 5

Presupuesto estimado 5

Lista de Interesados (stakeholders) 6

Requisitos de aprobación del proyecto 6

Asignación del Gerente de Proyecto y nivel de autoridad 6

Gerente de Proyecto 6

Niveles de autoridad 6

Personal y recursos preasignados 7

Aprobaciones 7

**Información del Proyecto**

**Datos**

| Empresa / Organización |  |
| --- | --- |
| Proyecto |  |
| Fecha de preparación | 02/09/2024 |
| Cliente | Publico General |
| Patrocinador principal |  |
| Gerente de Proyecto | Miguel Ravello |

**Patrocinador / Patrocinadores**

| **Nombre** | **Cargo** | **Departamento / División** | **Rama ejecutiva (Vicepresidencia)** |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  |  |

**Propósito y Justificación del Proyecto**

| Se busca innovar y marcar la diferencia de otras aplicaciones similares mediante el uso de recursos AI para generar las rutinas y resultados, acorde a los datos entregados por los usuarios. Todo esto para lograr salir de la generalidad y poder ofrecer un servicio lo más personalizado que sea posible ya que en un mundo lleno de opciones de entrenamiento, muchas personas luchan por encontrar una rutina adecuada, mantener la motivación y seguir progresando hacia sus metas. Las apps actuales pueden ser genéricas, difíciles de usar o no adaptarse a tus necesidades específicas. |
| --- |

**Descripción del Proyecto y Entregables**

| 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.    * La arquitectura de microservicios debe facilitar la escalabilidad horizontal y vertical. 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 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. |
| **Cronograma (Tiempo)** | |
| Gestión de Parámetros de Entrada | 16 días |
| Configuración y Seguimiento de Rutinas | 20 días |
| Visualización de Estadísticas | 17 días |
| Gestión de Alimentación | 24 días |
| **Costo** | |
| Gestión de Parámetros de Entrada |  |
| Configuración y Seguimiento de Rutinas |  |
| Visualización de Estadísticas |  |
| Gestión de Alimentación |  |
| **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. |

**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 | 24 sept 2024 |
| Configuración y Seguimiento de Rutinas | 15 oct 2024 |
| Visualización de Estadísticas | 1 nov 2024 |
| Gestión de Alimentación | 25 nov 2024 |

**Presupuesto estimado**

|  |
| --- |

**Lista de Interesados (stakeholders)**

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

**Requisitos de aprobación del proyecto**

|  |
| --- |

**Asignación del Gerente de Proyecto y nivel de autoridad**

**Gerente de Proyecto**

| **Nombre** | **Cargo** | **Departamento / División** |
| --- | --- | --- |
| **Miguel Ravello** | jefe de proyecto (Jp) | 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** |
| --- | --- | --- |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |