

**UNIVERSIDAD CONTINENTAL**

**FACULTAD DE INGENIERÍA**

**ESCUELA ACADÉMICO PROFESIONAL DE INGENIERÍA**

**DE SISTEMAS E INFORMÁTICA**



**PROYECTO**

**“Implementación de un aplicativo web para mejorar la calidad de vida de los oficinistas de la Universidad Continental”**

**PRESENTADO POR:**

APELLIDOS Y NOMBRES	PARTICIPACIÓN
Alcedo Muñoz Luis Fernando	100%
Inga Barzola Emil Hristo	100%
Paucarchuco Ramos Alexander David	50%
Ramirez Crispin William Alexis	100%
Nolasco Meza Nilton Anthony	100%
Navarro Arias Michael Brandon	100%
Tayne Huarocc Jeanpier	100%

**ASESOR:**

**Gamarra Moreno Job Daniel**

# HUANCAYO – PERÚ

2023

## LISTA DE CONTENIDO

PORTADA	1
LISTA DE CONTENIDO	2
LISTA DE TABLAS	7
LISTA DE FIGURAS	8
CAPÍTULO 1	9
PLANTEAMIENTO DEL ESTUDIO	9
1.1. Aspectos Generales de la Empresa	9
1.1.1. Organigrama	9
1.1.2. Misión y visión	9
1.2. Diagnóstico del Problema	9
1.3. Procesos de la Empresa	9
1.4. Oportunidad Encontrada	9
1.5. Detalles del Proyecto	9
CAPÍTULO 2	10
ESTUDIO DE FACTIBILIDAD	10
2.1. Alternativas de Solución	10
2.2. Factibilidad Técnica	10
2.2.1....	10
2.2.2....	10
2.3. Factibilidad Económica	10
2.3.1. Gastos generales	10
2.4. Factibilidad Operacional	10
2.4.1. Sistemas de ventas	10
CAPÍTULO 3	11

<b>ANÁLISIS DE REQUERIMIENTOS</b>	11
3.1. Metas del Sistema de Información	11
3.2. Requisitos del Sistema	11
3.2.1. Requerimientos funcionales	11
3.2.2. Requerimientos no funcionales	11
3.3. Identificación de Actores del Sistema	11
3.3.1....	11
<b>CAPÍTULO 4</b>	12
<b>PLANIFICACIÓN DEL PROYECTO</b>	12
4.1. Definición de Roles de Trabajo	12
4.1.1. Product owner	12
4.1.2. Scrum master	12
4.1.3. Team member	12
4.1.4. Tester	12
4.2. Product Backlog	12
4.3. Sprint Backlog	12
4.3.1. Sprint 1	12
4.3.2. Sprint 2	12
4.3.3. Sprint 3	12
4.3.4. Sprint 4	12
4.3.5. Sprint 5	13
4.4. Planificación de Sprints	13
4.4.1. Historias de usuario	13
4.4.2. Priorización de historias de usuario	13
4.5. Cronograma de Actividades	13
4.6. Gestión de Riesgos	13

CAPÍTULO 5	14
DISEÑO DEL SISTEMA DE INFORMACIÓN	14
5.1. Diseño de Diagramas UML	14
5.1.1. Diagramas de casos de uso	14
5.1.2. Diagramas de secuencia	14
5.1.3. Diagramas de colaboración	14
5.1.4. Diagramas de clases	14
5.2. Diseño de Base de Datos	14
5.2.1. Diseño conceptual (E/R)	14
5.2.2. Diseño lógico	14
5.2.3. Diseño físico	14
5.2.4. Modelado de base de datos	14
5.3. Diseño de Interfaces Básicas	14
5.3.1. Acceso login	15
5.3.2. Interfaz ...	15
CAPÍTULO 6	16
CODIFICACIÓN DEL SOFTWARE	16
6.1. Desarrollo del Sprint 1	16
6.1.1. Sprint planning	16
6.1.2. Sprint backlog	16
6.1.3. Historias de usuarios	16
6.1.4. Taskboard	16
6.1.5. Daily scrum	16
6.1.6. Sprint review	16
6.1.7. Criterios de aceptación	16
6.1.8. Resultados del sprint	16

6.1.8.1. Evidencias.	16
6.1.8.2. Prueba de desarrollo.	16
6.1.8.3. ....	17
6.1.9. Sprint retrospective	17
6.2. Desarrollo del Sprint 2	17
6.2.1. Sprint planning	17
6.2.2. Sprint backlog	17
6.2.3. Historias de usuarios	17
6.2.4. Taskboard	17
6.2.5. Daily scrum	17
6.2.6. Sprint review	17
6.2.7. Criterios de aceptación	17
6.2.8. Resultados del sprint	17
6.2.8.1. Evidencias.	17
6.2.8.2. Prueba de desarrollo.	18
6.2.8.3. ....	18
6.2.9. Sprint retrospective	18
6.3. Desarrollo del Sprint 3	18
6.3.1. Sprint planning	18
6.3.2. Sprint backlog	18
6.3.3. Historias de usuarios	18
6.3.4. Taskboard	18
6.3.5. Daily scrum	18
6.3.6. Sprint review	18
6.3.7. Criterios de aceptación	18
6.3.8. Resultados del sprint	18

6.3.8.1. Evidencias.	19
6.3.8.2. Prueba de desarrollo.	19
6.3.8.3. ....	19
6.3.9. Sprint retrospective	19
<b>6.4. Desarrollo del Sprint 4</b>	<b>19</b>
6.4.1. Sprint planning	19
6.4.2. Sprint backlog	19
6.4.3. Historias de usuarios	19
6.4.4. Taskboard	19
6.4.5. Daily scrum	19
6.4.6. Sprint review	19
6.4.7. Criterios de aceptación	19
6.4.8. Resultados del sprint	20
6.4.8.1. Evidencias.	20
6.4.8.2. Prueba de desarrollo.	20
6.4.8.3. ....	20
6.4.9. Sprint retrospective	20
<b>CAPÍTULO 7</b>	<b>21</b>
<b>PRUEBAS DE SOFTWARE</b>	<b>21</b>
7.1. Plan de Pruebas	21
CONCLUSIONES	22
RECOMENDACIONES	23
ANEXOS	24
Anexo 01. Manual Técnico	25
Anexo 02. Manual de Usuario	26

## **CAPÍTULO 1**

### **PLANTEAMIENTO DEL ESTUDIO**

#### **1.1. Aspectos Generales de la Empresa**

La Universidad Continental es una institución privada ubicada en Huancayo, Perú, con filiales en Lima, Arequipa, Cusco y Florida (EE.UU). Fundada el 12 de junio de 1998, la universidad ha crecido significativamente y ahora cuenta con más de 60,000 estudiantes.

#### **FACULTADES Y CARRERAS**

La universidad ofrece una amplia gama de programas académicos a través de sus cinco facultades, que incluyen 28 carreras profesionales. Además, cuenta con programas de posgrado y especialización, diseñados para satisfacer las necesidades del mercado laboral actual.

#### **MODALIDADES DE ESTUDIO**

La Universidad Continental se destaca por su flexibilidad en las modalidades de estudio, ofreciendo opciones presenciales, semipresenciales y a distancia. Esto permite a los estudiantes adaptar su educación a sus necesidades personales y profesionales.

#### **ENFOQUE EDUCATIVO**

La universidad promueve un modelo educativo centrado en el estudiante, con metodologías de aprendizaje experiencial, experimental y colaborativo. Este enfoque

prepara a los estudiantes para enfrentar situaciones reales en un mundo en constante cambio.

## **INTERNACIONALIZACIÓN**

La Universidad Continental impulsa la internacionalización a través de más de 290 convenios internacionales y la posibilidad de obtener un doble grado académico en EE.UU. Esto permite a los estudiantes adquirir una formación global y mejorar su empleabilidad.

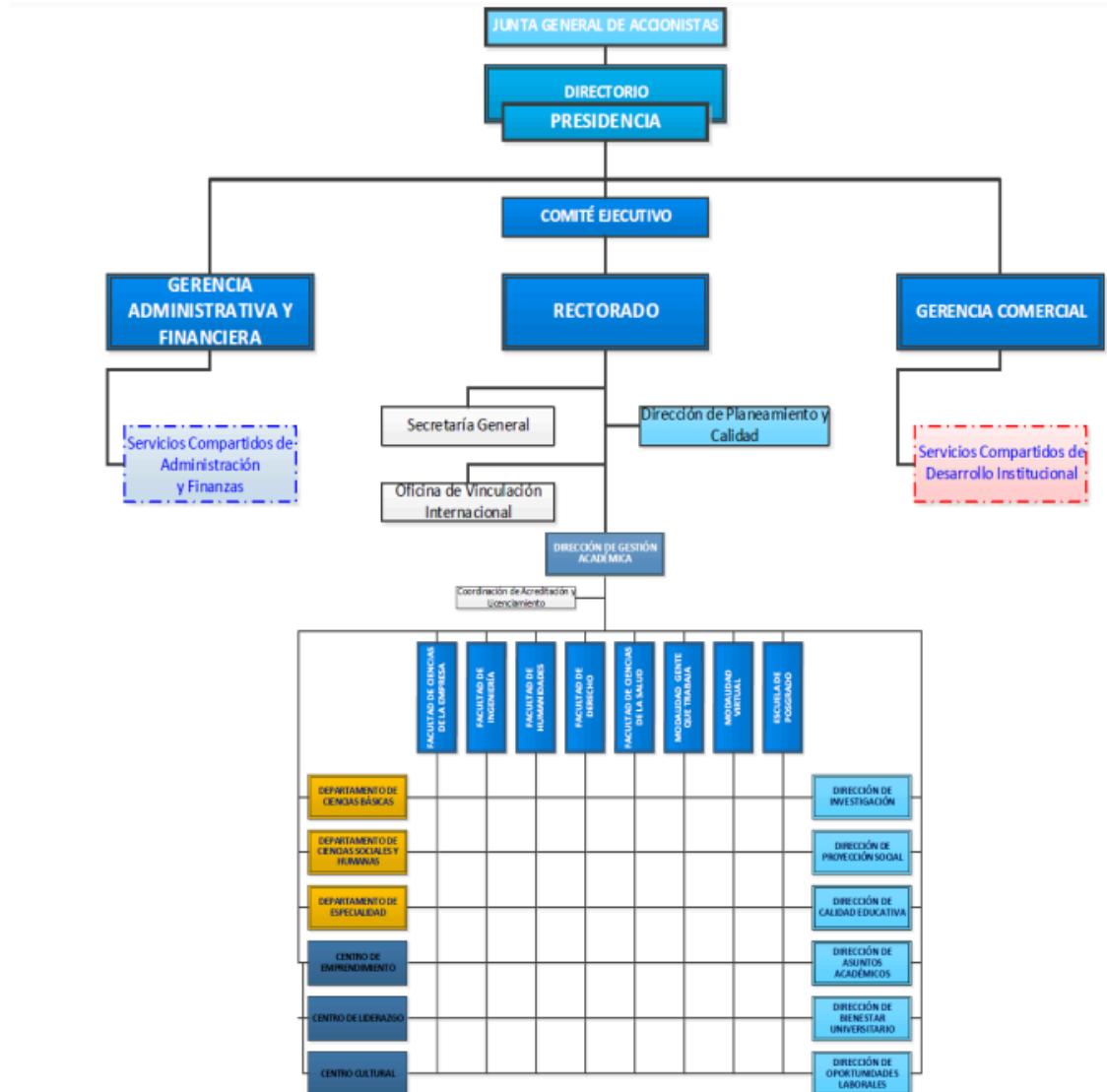
## **INFRAESTRUCTURA Y RECURSOS**

La universidad cuenta con modernos campus equipados con laboratorios de última generación, espacios de aprendizaje colaborativo y tecnología innovadora. Además, ofrece programas de apoyo como el Growth Center Continental y la incubadora de negocios Wichay.

## **COMPROMISO CON LA CALIDAD**

La Universidad Continental ha obtenido diversas acreditaciones y certificaciones internacionales, reafirmando su compromiso con la excelencia académica y la calidad educativa.

### **1.1.1. Organigrama**



## **Figura N° 1 Organigrama de la Universidad Continental**

### **1.1.2. Misión y visión**

## **Misión :**

Somos una organización de educación superior que conecta personas e ideas para impulsar la innovación y el bienestar integral a través de una cultura de pensamiento y acción emprendedora, la investigación y el impacto social.

## **Visión :**

Ser la mejor organización de educación superior posible que brinda oportunidades para el aprendizaje, la investigación y que une personas e ideas que buscan hacer realidad sueños y aspiraciones de prosperidad en un entorno incierto.

## **1.2. Diagnóstico del Problema**

El diagnóstico del problema de la alta carga laboral en los oficinistas de la Universidad Continental se basa en la identificación y análisis de varios factores clave que afectan su calidad de vida. A continuación, se detallan estos factores:

- **EVALUACIÓN DEL ESTRÉS LABORAL:** A través de encuestas y entrevistas, se ha identificado que un alto porcentaje de oficinistas experimenta niveles significativos de estrés debido a la acumulación de tareas y responsabilidades. Este estrés se manifiesta en síntomas físicos y emocionales, como fatiga, irritabilidad y problemas de concentración.
- **ANÁLISIS DEL TIEMPO DE TRABAJO:** Los registros de horas trabajadas muestran que muchos oficinistas exceden regularmente las horas laborales estándar, lo que reduce su tiempo disponible para actividades personales y familiares. Este desequilibrio entre la vida laboral y personal contribuye a una disminución en la satisfacción y motivación laboral.
- **RIESGOS PSICOSOCIALES:** Se ha observado un aumento en los casos de ansiedad y depresión entre los oficinistas, lo cual está directamente relacionado con el estrés laboral constante. Estos problemas de salud mental no solo afectan el bienestar de los empleados, sino también su productividad y eficiencia en el trabajo.

- **ENTORNO DE TRABAJO:** La falta de recursos adecuados y el ambiente de trabajo poco favorable también contribuyen a la alta carga laboral. La falta de herramientas eficientes para la gestión de tareas y la comunicación interna deficiente agravan el problema.

El diagnóstico revela que la alta carga laboral es un problema multifacético que requiere una intervención integral. La implementación de un aplicativo web puede ser una solución efectiva para abordar estos factores, mejorando así la calidad de vida de los oficinistas.

### 1.3. Procesos de la Empresa



#### **1.4. Oportunidad Encontrada**

La implementación de un aplicativo web para mejorar la calidad de vida de los oficinistas de la Universidad Continental presenta varias oportunidades significativas:

**MEJORA EN LA GESTIÓN DEL TIEMPO:** Un aplicativo web puede ayudar a los oficinistas a organizar y priorizar sus tareas de manera más eficiente, reduciendo el estrés asociado con la alta carga laboral y mejorando su productividad.

**FOMENTO DEL BIENESTAR:** Al incluir recursos y programas de bienestar, como ejercicios de relajación y apoyo psicológico, el aplicativo puede contribuir a una mejor salud mental y física de los empleados, reduciendo los riesgos psicosociales.

**OPTIMIZACIÓN DE LA COMUNICACIÓN:** Herramientas de comunicación interna más efectivas pueden facilitar la colaboración y reducir el tiempo dedicado a reuniones innecesarias, permitiendo a los oficinistas enfocarse en tareas más importantes.

**EQUILIBRIO ENTRE VIDA LABORAL Y PERSONAL:** Al gestionar mejor las tareas y reducir el estrés, los oficinistas pueden tener más tiempo y energía para dedicar a sus actividades personales y familiares, mejorando su equilibrio entre la vida laboral y personal.

**AUMENTO DE LA SATISFACCIÓN Y MOTIVACIÓN LABORAL:** Un entorno de trabajo más saludable y organizado puede aumentar la satisfacción y motivación de los empleados, lo que a su vez puede mejorar la retención de talento y la eficiencia operativa de la universidad.

**INNOVACIÓN Y COMPETITIVIDAD:** La adopción de tecnologías innovadoras como un aplicativo web demuestra el compromiso de la Universidad Continental con la modernización y la mejora continua, posicionándose como una institución competitiva y atractiva tanto para empleados como para estudiantes.

### **1.5. Detalles del Proyecto**

El proyecto se enfoca en el desarrollo de un aplicativo web integral que optimice la gestión de tareas y recursos para los oficinistas de la Universidad Continental. Este aplicativo mejorará la calidad de vida de los empleados al facilitar la organización de sus actividades diarias, reducir el estrés laboral y promover un equilibrio saludable entre la vida laboral y personal.

## **ALCANCE**

### **DENTRO DEL ALCANCE:**

- Desarrollo de un módulo de gestión de tareas para la organización y priorización de actividades.
- Implementación de herramientas de comunicación interna para mejorar la colaboración entre equipos.
- Integración de recursos de bienestar, como ejercicios de relajación
- Diseño de una interfaz de usuario intuitiva y amigable para facilitar su uso por parte de los oficinistas.

- Realización de pruebas exhaustivas para garantizar el correcto funcionamiento del aplicativo.

#### **FUERA DEL ALCANCE:**

- Gestión de citas o agendamiento de reuniones externas.
- Integración con sistemas de terceros no relacionados con la gestión interna de la universidad.
- Desarrollo de versión móvil para la gestión de tareas.

#### **OBJETIVO GENERAL**

Desarrollar un aplicativo web integral para optimizar la gestión de tareas y recursos de los oficinistas de la Universidad Continental, mejorando su eficiencia, bienestar y satisfacción laboral.

#### **OBJETIVOS ESPECÍFICOS**

- Reducir el tiempo dedicado por los oficinistas a la gestión manual de tareas.
- Automatizar tareas repetitivas, liberando tiempo para actividades más importantes.
- Agilizar el proceso de comunicación interna.
- Eliminar errores humanos en la organización de tareas.
- Proporcionar herramientas de bienestar para reducir el estrés laboral.
- Mejorar la comunicación y colaboración entre equipos.
- Aumentar la satisfacción y motivación laboral de los oficinistas.
- Facilitar el equilibrio entre la vida laboral y personal.
- Contribuir a un entorno de trabajo más saludable y productivo.
- Asegurar el cumplimiento de normas y regulaciones relacionadas con la gestión de recursos humanos.
- Fortalecer la imagen de la Universidad Continental como una institución moderna y eficiente.

## ENTREGABLES

- Aplicativo web desarrollado .
- Documentación completa del aplicativo, incluyendo manuales de usuario, guías técnicas y especificaciones de diseño.
- Pruebas unitarias y de integración que confirmen el correcto funcionamiento del aplicativo.
- Capacitación al personal de la universidad en el uso del aplicativo web.

Item	Historia de Usuario	Nombre de tarea	Duración	Start	End	Predecesores
Configuración y Gestión de Usuarios	Gestionar cuenta de usuarios	Implementar creación de cuenta	1/2 Dia	21/8/2024	21/8/2024	
Configuración y Gestión de Usuarios		Crear opción para editar cuenta	1/2 Dia	21/8/2024	21/8/2024	Implementar visualización de cuenta
Configuración y Gestión de Usuarios		Implementar eliminación de cuenta	1/2 Dia	22/8/2024	22/8/2024	Crear opción para editar cuenta
Configuración y Gestión de Usuarios	Poder iniciar sesión en el sistema	Implementar autenticación de usuarios	1/2 Dia	22/8/2024	22/8/2024	Implementar eliminación de cuenta
Configuración y Gestión de Usuarios		Crear opción para iniciar sesión	1/2 Dia	22/8/2024	22/8/2024	Implementar autenticación de usuarios
Configuración y Gestión de Usuarios		Implementar sesión de usuario persistente	1/2 dia	23/8/2024	23/8/2024	Crear recuperación de contraseña
Configuración y Gestión de Usuarios	Tener opción de recuperar contraseña	Implementar interfaz para recuperación de contraseña	1 dia	24/8/2024	24/8/2024	Implementar sesión de usuario persistente
Configuración y Gestión de Usuarios		Configurar opciones de correo electrónicos para recuperación	1 dia	25/8/2024	25/8/2024	Implementar interfaz para recuperación de contraseña
Configuración y Gestión de Usuarios		Validar proceso de recuperación	1 dia	26/8/2024	26/8/2024	Configurar envío de correos electrónicos para recuperación
Configuración y Gestión de Usuarios	Suspender cuentas de usuarios inactivos	Implementar suspensión de cuentas	2 dias	27/8/2024	28/8/2024	Validar proceso de recuperación
Configuración y Gestión de Usuarios		Crear notificación para usuarios suspendidos	1 dia	29/8/2024	29/8/2024	Implementar suspensión de cuenta
Configuración y Gestión de Usuarios	Modificar datos de los usuarios registrados	Implementar actualización de datos de usuario	2 dias	1/1/2024	1/1/2024	Implementar eliminación de cuenta
Configuración y Gestión de Usuarios		Validar datos ingresados	1 dia	2/1/2024	2/1/2024	Implementar edición de datos de usuario
Configuración y Gestión de Usuarios	Actualizar mis datos personales	Implementar interfaz para actualización de datos	2 dias	3/1/2024	4/1/2024	Implementar interfaz para actualización de datos
Configuración y Gestión de Usuarios		Validar datos ingresados	1/2 dia	4/1/2024	4/1/2024	Validar datos ingresados
Configuración y Gestión de Usuarios	Consultar historial de cambios	Notificar cambios realizados en los datos	1/2 dia	4/1/2024	4/1/2024	Validar datos ingresados
Registro y Gestión de Clientes	Registrar nuevos clientes	Implementar formulario de registro de clientes	2 dias	5/9/2024	6/9/2024	Implementar interfaz para actualización de datos
Registro y Gestión de Clientes		Validar datos ingresados	2 dias	7/9/2024	8/9/2024	Implementar formulario de registro de clientes
Registro y Gestión de Clientes		Enviar notificación de registro	2 dias	9/9/2024	10/9/2024	Validar datos ingresados
Registro y Gestión de Clientes	Actualizar datos de clientes	Implementar interfaz para actualización de datos de clientes	2 dias	11/9/2024	12/9/2024	Enviar notificación de registro
Registro y Gestión de Clientes		Validar datos ingresados	2 dias	13/9/2024	14/9/2024	Implementar interfaz para actualización de datos de clientes
Registro y Gestión de Clientes		Crear interfaz para selección de rutinas	2 dias	15/9/2024	16/9/2024	Validar datos ingresados
Registro y Gestión de Clientes	Revisar el progreso de los clientes	Notificar a los clientes sobre sus rutinas asignadas	1 dia	16/9/2024	17/9/2024	Implementar asignación de rutinas
Registro y Gestión de Clientes		Implementar seguimiento del progreso de clientes	2 dias	26/9/2024	27/9/2024	Notificar a los clientes sobre sus rutinas asignadas
Registro y Gestión de Clientes		Generar informes de progreso	2 dias	28/9/2024	29/9/2024	Implementar seguimiento del progreso de clientes
Registro y Gestión de Clientes	Asignar rutinas de ejercicio a clientes	Configurar alertas de progreso	1 dia	30/9/2024	30/9/2024	Generar informes de progreso
Registro y Gestión de Clientes		Implementar asignación de rutinas	2 dias	31/9/2024	3/10/2024	Generar informes de progreso
Registro y Gestión de Clientes	Programar y gestionar citas con entrenadores	Implementar visualización de metas	2 dias	4/10/2024	5/10/2024	Generar informes de progreso
Registro y Gestión de Clientes		Enviar confirmaciones de citas	1 dia	6/10/2024	6/10/2024	Implementar programación de citas
Registro y Gestión de Clientes		Implementar gestión de citas	1 dia	4/10/2024	4/10/2024	Enviar confirmaciones de citas
Rutinas de Ejercicio y Seguimiento	Crear nuevas rutinas de ejercicio	Implementar creación de rutinas	2 dias	19/9/2024	20/9/2024	Implementar creación de rutinas
Rutinas de Ejercicio y Seguimiento		Validar datos ingresados	2 dias	21/9/2024	22/9/2024	Validar datos ingresados
Rutinas de Ejercicio y Seguimiento	Consultar historial de rutinas y ejercicios	Implementar configuración de cambios	2 dias	23/9/2024	24/9/2024	Validar datos ingresados
Rutinas de Ejercicio y Seguimiento		Implementar visualización de cambios	2 dias	25/9/2024	26/9/2024	Implementar configuración de cambios
Rutinas de Ejercicio y Seguimiento	Consultar y ajustar metas de entrenamiento	Implementar filtro de búsqueda	2 dias	19/9/2024	20/9/2024	Implementar visualización del historial
Rutinas de Ejercicio y Seguimiento		Implementar asignación de rutinas	2 dias	21/9/2024	22/9/2024	Implementar filtro de búsqueda
Rutinas de Ejercicio y Seguimiento	Programar y gestionar citas con entrenadores	Implementar visualización de metas	2 dias	23/9/2024	24/9/2024	Implementar asignación de rutinas
Rutinas de Ejercicio y Seguimiento		Enviar confirmaciones de citas	1 dia	3/10/2024	3/10/2024	Implementar visualización de metas
Rutinas de Ejercicio y Seguimiento	Realizar seguimiento de las citas de entrenamiento	Implementar gestión de citas	2 dias	4/10/2024	5/10/2024	Enviar confirmaciones de citas
Rutinas de Ejercicio y Seguimiento		Validar datos ingresados	1 dia	6/10/2024	6/10/2024	Validar datos ingresados
Rutinas de Ejercicio y Seguimiento		Notificar sobre actualizaciones	1 dia	7/10/2024	7/10/2024	Validar datos ingresados
Rutinas de Ejercicio y Seguimiento		Implementar visualización de recomendaciones de nutrición	2 dias	10/10/2024	11/10/2024	Validar datos ingresados
Nutrición y Planificación Alimentaria	Consultar recomendaciones de nutrición	Validar rutinas creadas	2 dias	21/9/2024	23/9/2024	Implementar creación de rutinas
Nutrición y Planificación Alimentaria		Generar plantillas de rutinas	1 dia	23/9/2024	23/9/2024	Validar rutinas creadas
Nutrición y Planificación Alimentaria	Programar sesiones de entrenamiento	Implementar programación de sesiones	2 dias	24/9/2024	25/9/2024	Generar plantillas de rutinas
Nutrición y Planificación Alimentaria		Crear interfaz para visualizar sesiones	1 dia	26/9/2024	26/9/2024	Implementar programación de sesiones
Nutrición y Planificación Alimentaria	Consultar y ajustar metas de entrenamiento	Validar plantillas de rutinas	2 dias	27/9/2024	27/9/2024	Crear interfaz para visualizar sesiones
Nutrición y Planificación Alimentaria		Implementar visualización de metas	2 dias	28/9/2024	29/9/2024	Validar plantillas de rutinas
Nutrición y Planificación Alimentaria	Consultar y ajustar metas de entrenamiento	Crear opción para ajustar metas	2 dias	30/9/2024	1/10/2024	Implementar visualización de metas
Nutrición y Planificación Alimentaria		Generar informes de progreso	2 dias	2/10/2024	3/10/2024	Crear opción para ajustar metas
Nutrición y Planificación Alimentaria	Realizar seguimiento de las citas de entrenamiento	Implementar seguimiento de citas	2 dias	4/10/2024	5/10/2024	Generar informes de progreso
Nutrición y Planificación Alimentaria		Validar datos ingresados	1 dia	6/10/2024	6/10/2024	Actualizar datos de clientes
Nutrición y Planificación Alimentaria		Notificar sobre actualizaciones	1 dia	7/10/2024	7/10/2024	Actualizar datos de clientes
Nutrición y Planificación Alimentaria	Consultar recomendaciones de nutrición	Implementar visualización de recomendaciones de nutrición	2 dias	10/10/2024	11/10/2024	Actualizar datos de clientes
Nutrición y Planificación Alimentaria		Crear filtros para recomendaciones	2 dias	12/10/2024	13/10/2024	Implementar visualización de recomendaciones de nutrición
Nutrición y Planificación Alimentaria	Crear nuevos planes de alimentación	Implementar creación de planes de alimentación	2 dias	14/10/2024	15/10/2024	Crear filtros para recomendaciones
Nutrición y Planificación Alimentaria		Validar plantillas de planes	2 dias	17/10/2024	18/10/2024	Implementar creación de planes de alimentación
Nutrición y Planificación Alimentaria	Revise el progreso de los clientes en cuanto a nutrición	Implementar seguimiento del progreso en nutrición	2 dias	19/10/2024	20/10/2024	Validar plantillas de planes
Nutrición y Planificación Alimentaria		Generar informes de nutrición	2 dias	21/10/2024	22/10/2024	Implementar seguimiento del progreso en nutrición
Nutrición y Planificación Alimentaria	Programar consultas de nutrición	Implementar programación de consultas	2 dias	23/10/2024	24/10/2024	Generar informes de nutrición
Nutrición y Planificación Alimentaria		Enviar confirmaciones de consultas	1 dia	25/10/2024	26/10/2024	Implementar programación de consultas
Nutrición y Planificación Alimentaria	Consultar y ajustar metas de nutrición	Implementar ajustes en consultas	1 dia	26/10/2024	26/10/2024	Enviar confirmaciones de consultas
Nutrición y Planificación Alimentaria		Implementar visualización de metas	2 dias	27/10/2024	28/10/2024	Implementar ajustes en consultas
Nutrición y Planificación Alimentaria	Realizar seguimiento de las citas de nutrición	Crear opción para ajustar metas	2 dias	29/10/2024	30/10/2024	Implementar visualización de metas
Nutrición y Planificación Alimentaria		Generar informes de seguimiento	2 dias	31/10/2024	1/11/2024	Implementar visualización de metas
Fisioterapia y Seguimiento	Crear nuevo tratamiento de fisioterapia	Implementar creación de tratamientos	2 dias	2/11/2024	3/11/2024	Implementar seguimiento de citas
Fisioterapia y Seguimiento		Validar tratamientos creados	2 dias	7/11/2024	8/11/2024	Implementar creación de tratamientos
Fisioterapia y Seguimiento	Consultar y ajustar metas de fisioterapia	Generar plantillas de tratamientos	1 dia	9/11/2024	10/11/2024	Validar tratamientos creados
Fisioterapia y Seguimiento		Implementar seguimiento de tratamientos	2 dias	12/11/2024	13/11/2024	Generar plantillas de tratamientos
Fisioterapia y Seguimiento	Realizar seguimiento de las citas de fisioterapia	Implementar seguimiento de citas	2 dias	24/11/2024	25/11/2024	Implementar seguimiento de tratamientos
Fisioterapia y Seguimiento		Crear informes de seguimiento	2 dias	26/11/2024	27/11/2024	Implementar seguimiento de citas
Pagos, Facturación y Optimización	Gestionar pagos y facturación	Implementar creación de pagos	2 dias	28/11/2024	29/11/2024	Crear informes de seguimiento
Pagos, Facturación y Optimización		Validar datos de pago y facturación	2 dias	30/11/2024	31/11/2024	Implementar creación de pagos
Pagos, Facturación y Optimización	Realizar pagos y consultar facturación	Implementar interfaz para realizar pagos	2 dias	3/12/2024	4/12/2024	Validar datos de pago y facturación
Pagos, Facturación y Optimización		Generar informes de facturación	2 dias	5/12/2024	6/12/2024	Implementar interfaz para realizar pagos
Pagos, Facturación y Optimización	Revisar el progreso de los clientes	Implementar seguimiento del progreso de clientes	2 dias	1/1/2024	2/12/2024	Generar informes de facturación
Pagos, Facturación y Optimización		Generar informes de progreso	2 dias	3/12/2024	4/12/2024	Implementar seguimiento del progreso de clientes
Pagos, Facturación y Optimización	Programar citas para clientes	Implementar programación de citas	2 dias	1/2/2024	2/12/2024	Generar informes de progreso
Pagos, Facturación y Optimización		Enviar confirmaciones de citas	1 dia	3/12/2024	3/12/2024	Implementar programación de citas
Pagos, Facturación y Optimización		Optimización de la gestión de citas	3 dias	4/12/2024	6/12/2024	Enviar confirmaciones de citas

## CAPÍTULO 2

### ESTUDIO DE FACTIBILIDAD

#### **2.1. Alternativas de Solución**

Implementar una plataforma web innovadora que fomente el bienestar físico de los oficinistas en la Universidad Continental, basada en tecnologías modernas como JavaScript para la lógica del cliente, Angular para una arquitectura web escalable y eficiente, y MongoDB para el almacenamiento seguro y ágil de datos como rutinas personalizadas, históricas de ejercicios y progresos individuales. La plataforma integrará un sistema de gestión de rutinas adaptables, un motor de seguimiento del progreso y recomendaciones inteligentes personalizadas, todo centralizado en un dashboard interactivo con visualizaciones gráficas detalladas. Además, se utilizarán metodologías como Git Flow para garantizar un desarrollo ordenado y Docker para facilitar el despliegue en distintos entornos. Esta solución no solo optimiza la calidad de vida y fomenta hábitos saludables en los oficinistas, sino que también asegura un sistema robusto, escalable y eficiente que responde a las necesidades del entorno laboral moderno.

#### **2.2. Factibilidad Técnica**

##### **2.2.1. Disponibilidad de Recursos Tecnológicos:**

**Visual Studio Code:** Es un editor de código fuente gratuito y extensible desarrollado por Microsoft, conocido por su integración con una variedad de 14 lenguajes y herramientas de desarrollo, proporcionando un entorno de desarrollo versátil y potente.

**JavaScript:** Es un lenguaje de programación soportado por todos los navegadores modernos y cuenta con muchas bibliotecas y frameworks que facilitan el desarrollo web.

**React:** Es un framework popular para desarrollar aplicaciones web robustas y escalables, con una documentación completa y una comunidad activa que proporciona soporte y recursos.

**MongoDB:** Es una base de datos NoSQL orientada a documentos, conocida por su flexibilidad y escalabilidad, ideal para manejar grandes volúmenes de datos no estructurados o semiestructurados.

**Docker:** Es una plataforma de contenerización que permite crear, desplegar y ejecutar aplicaciones en contenedores, asegurando consistencia entre entornos y facilitando la implementación y escalabilidad.

### 2.2.2. Requerimientos de Hardware

Requisitos Mínimos	Requisitos Recomendados
Procesador: Intel Core i5 o Ryzen 5 mínimo 6 núcleos y 12 hilos, eficiente para un buen desarrollo.	Procesador: Intel Core i7 o Ryzen 7 que ofrecen excelente multitarea y rendimiento para compilaciones y ejecuciones eficientes
Memoria RAM: 16 GB, suficiente para operar varias aplicaciones y servicios a la vez, incluidos servidores de desarrollo y bases de datos.	Memoria RAM: 32 GB, permitiendo trabajar con múltiples proyectos y contenedores sin problemas de rendimiento.
Almacenamiento: SSD de 512GB, que proporciona tiempos de carga rápidos y espacio suficiente para proyectos y datos.	Almacenamiento: SSD NVMe de 1 TB, que proporciona tiempos de carga rápidos y una gran capacidad de almacenamiento.

Tarjeta Gráfica: NVIDIA GeForce GTX 1650 o similar. Aunque el desarrollo web no exige mucho en términos gráficos.

Sistema Operativo: Windows 10/11, Linux (Ubuntu, Fedora), o macOS, asegurando flexibilidad y compatibilidad con Docker y otras herramientas de desarrollo

### **2.2.3. Seguridad**

- JavaScript y Angular proporcionan mecanismos para implementar prácticas de seguridad robustas en el desarrollo de la aplicación, incluyendo autenticación, autorización y protección contra ataques comunes como Cross-Site Scripting (XSS).
- MongoDB permiten implementar cifrado de datos y medidas de seguridad avanzadas para proteger la información sensible de los pacientes.
- Docker mejora la seguridad al aislar las aplicaciones en contenedores, reduciendo el riesgo de conflictos y vulnerabilidades entre servicios.

### **2.2.4. Disponibilidad de Conocimiento y Soporte**

La comunidad y la documentación extensiva de Visual Studio Code, JavaScript, Angular, MongoDB y Docker aseguran que el equipo de desarrollo pueda encontrar soporte y soluciones a problemas técnicos durante el desarrollo del proyecto

## **2.3. Factibilidad Económica**

### **2.3.1. Gastos generales**

**Costos de Personal:**

**Desarrolladores:** 8 desarrolladores x S/. 8/hora/desarrollador x 40 horas/semana x 9 semanas = **S/. 23,040**

**Tester:** 1 tester x S/. 10/hora/tester x 40 horas/semana x 9 semanas = **S/. 3,600**

**Diseñador UI/UX:** 1 diseñador x S/. 7/hora/diseñador x 40 horas/semana x 9 semanas = **S/. 2,016**

**Total Costos de Personal:**

**S/. 23,040 + S/. 3,600 + S/. 2,016 = S/. 28,656**

**Costos de Software y Licencias:**

- Entorno de desarrollo integrado (IDE): S/. 200
- Herramientas de control de versiones: S/. 100
- Herramientas de testing: S/. 150
- Licencias de software: S/. 300

**Total Costos de Software y Licencias: S/. 750**

**Costo Total del Proyecto:**

**Costos de Personal + Costos de Software y Licencias + Otros Gastos = Costo**

**Total del Proyecto**

**S/. 28,656 + S/. 750 = S/. 29,406**

## **2.4. Factibilidad Operacional**

### **2.4.1. Capacitación y competencias del personal:**

- **Entrenamiento del personal:** Es esencial formar a los administrativos, docentes y estudiantes en el uso adecuado del sistema web.
- **Recursos de formación continua:** Se deben establecer programas de capacitación constante para actualizar las competencias del personal a medida que el sistema evolucione.

### **2.4.2. Adaptación al entorno laboral:**

- **Compatibilidad con la infraestructura actual:** El sistema debe integrarse con la infraestructura tecnológica existente, aprovechando herramientas como Docker para facilitar su implementación.
- **Soporte técnico:** Es necesario contar con un soporte de TI eficiente para resolver problemas operativos de forma rápida y garantizar la continuidad del servicio.

### **2.4.3. Transformación de procesos operativos:**

- **De manual a automatizado:** La transición de procesos manuales a automáticos requiere un periodo de adaptación cuidadosamente planificado para minimizar resistencias y favorecer su aceptación.
- **Mejoras en los procesos:** Los procedimientos actuales deben revisarse y ajustarse para aprovechar al máximo las capacidades del nuevo sistema, incrementando la eficiencia operativa.

### **2.4.4. Mantenimiento sostenible a largo plazo:**

- **Actualizaciones y soporte técnico:** El sistema necesitará mantenimiento regular y actualizaciones para garantizar su funcionalidad y seguridad, utilizando tecnologías robustas como JavaScript, Angular, MySQL, MongoDB y Docker.

#### **2.4.5. Evaluación constante del sistema:**

- **Retroalimentación y mejoras:** Establecer mecanismos de recolección continua de opiniones permitirá implementar ajustes y mejoras, asegurando que el sistema siga siendo efectivo y pertinente para las necesidades de la facultad.

## CAPÍTULO 3

### ANÁLISIS DE REQUERIMIENTOS

#### **3.1. Metas del Sistema de Información**

El sistema de información que se desarrollará tiene como objetivo principal optimizar el bienestar físico de los oficinistas de la Universidad Continental mediante la implementación de una plataforma web especializada. Las metas específicas que se persiguen con este sistema son las siguientes:

##### **3.1.1. Registro y Perfil de Usuario**

Esta funcionalidad permite a los oficinistas crear y gestionar perfiles personalizados que incluyan metas de salud y preferencias individuales.

**Descripción:** Diseñar un sistema de registro que permita la configuración inicial de perfiles y la actualización de datos personales y de progreso.

**Justificación:** Facilitar la personalización y adaptación del sistema a las necesidades específicas de cada usuario, fomentando una experiencia más enriquecedora.

##### **3.1.2. Gestión de Rutinas Personalizadas**

Facilita la creación, personalización y seguimiento de rutinas de ejercicio alineadas con los objetivos de los usuarios.

**Descripción:** Implementar un módulo para diseñar y modificar rutinas según las condiciones físicas y metas individuales.

**Justificación:** Proveer un enfoque estructurado y flexible para promover el bienestar físico, optimizando el tiempo y recursos de los oficinistas.

##### **3.1.3. Seguimiento de Progreso**

Proveer herramientas para registrar y visualizar el progreso físico en términos de rendimiento, bienestar general y metas cumplidas.

**Descripción:** Crear un sistema interactivo que permita al usuario monitorear y analizar su evolución mediante gráficos y estadísticas.

**Justificación:** Incrementar la motivación y el compromiso con hábitos saludables mediante la visualización del avance personal.

#### **3.1.4. Recomendaciones Personalizadas**

Automatizar la generación de sugerencias basadas en el historial y las metas del usuario.

**Descripción:** Desarrollar un motor de recomendaciones que ajuste las rutinas de ejercicio y planes de alimentación en tiempo real.

**Justificación:** Maximizar la efectividad de las actividades propuestas y garantizar la coherencia con los objetivos de bienestar del usuario.

#### **3.1.5. Planificación Alimenticia Personalizada**

Diseñar planes de alimentación que complementen las rutinas físicas y se adapten a las necesidades nutricionales.

**Descripción:** Incorporar un módulo para crear, ajustar y visualizar planes alimenticios integrados en el dashboard del usuario.

**Justificación:** Fomentar un enfoque integral al bienestar físico y mental mediante una alimentación equilibrada.

#### **3.1.6. Generación de Informes y Acceso a Historiales**

Reducir la carga manual al generar informes detallados y mantener un historial completo de rutinas y progresos.

**Descripción:** Implementar una funcionalidad que permita la generación automática de informes en formato PDF, accesibles desde el perfil del usuario.

**Justificación:** Facilitar la toma de decisiones informadas y la planificación de metas futuras mediante un acceso ordenado y visual a los datos.

### **3.1.7. Seguridad y Accesibilidad**

Garantizar la protección de los datos personales y ofrecer una experiencia accesible para toda la comunidad universitaria.

**Descripción:** Incorporar medidas de seguridad robustas y una interfaz que sea compatible con diferentes dispositivos.

**Justificación:** Asegurar la privacidad de los usuarios y fomentar la adopción masiva del sistema gracias a su facilidad de uso.

### **3.1.8. Fomento de la Cultura de Bienestar**

Promover la participación activa en actividades saludables mediante la integración de eventos y comunicación de logros.

**Descripción:** Crear un espacio en la plataforma para notificar eventos relacionados con el bienestar y mostrar reconocimientos por logros alcanzados.

**Justificación:** Incrementar la motivación y el sentido de comunidad al destacar las actividades y los logros individuales o grupales.

## **3.2. Requisitos del Sistema**

### **3.2.1. Requerimientos funcionales**

Los requerimientos funcionales detallan las funcionalidades que el sistema debe ofrecer para satisfacer las necesidades del cliente, describiendo lo que debe hacer el sistema para cumplir con los objetivos y expectativas establecidos.

ID	Requerimiento	Descripción
RF001	Registro de Usuarios	Permitir el registro de nuevos usuarios con datos personales, metas de salud, nivel de condición física inicial.
RF002	Gestión de Perfiles	Habilitar la edición de datos personales, actualización de metas, visualización del progreso acumulado y configuración de preferencias en el perfil del usuario.
RF003	Creación de Rutinas de Ejercicio	Proporcionar herramientas para diseñar rutinas de ejercicio personalizadas basadas en los niveles de condición física y metas individuales, ajustables según el progreso.
RF004	Biblioteca de Rutinas	Ofrecer una lista de rutinas predefinidas categorizadas (ejemplo: fuerza, cardio, estiramientos) que puedan seleccionarse y personalizarse por el usuario.
RF005	Seguimiento del Progreso	Implementar un módulo para registrar y analizar el rendimiento físico, cambios corporales y metas alcanzadas, presentado en gráficos interactivos y estadísticas.
RF006	Recomendaciones de Ajustes en Rutinas	Incorporar un motor que sugiera modificaciones en las rutinas de ejercicio basadas en el desempeño del usuario y las metas establecidas.
RF007	Configuración de Planes Alimenticios	Permitir la creación de planes alimenticios personalizados que complementen las rutinas de ejercicio y sean ajustables según los hábitos, horarios y metas nutricionales.
RF008	Generación de Informes Personalizados	Facilitar la generación automática de informes descargables en PDF que incluyan progreso físico, ajustes recomendados y planes de acción.
RF009	Historial de Progreso	Habilitar un almacenamiento centralizado y accesible del historial de rutinas, progresos y planes personalizados para cada usuario.
RF010	Seguridad de Datos	Implementar medidas de cifrado para proteger la información personal y de progreso, junto con autenticación basada en roles para limitar el acceso según los permisos.

RF011	Compatibilidad Multidispositivo	Diseñar la plataforma para que sea completamente accesible desde dispositivos móviles, tabletas y computadoras con navegadores populares.
RF012	Notificaciones Personalizadas	Proporcionar notificaciones en tiempo real sobre eventos, logros alcanzados, recordatorios de rutinas y sugerencias de mejora.
RF013	Dashboard Interactivo	Crear un panel principal que consolide y visualice las métricas clave del usuario (progreso, recomendaciones, rutinas activas) de forma gráfica e intuitiva.
RF014	Integración con Módulos Complementarios	Permitir la integración con módulos adicionales en el futuro, como seguimiento fisioterapéutico o bienestar mental, para ampliar las funcionalidades.
RF015	Capacitación y Soporte a Usuarios	Incluir tutoriales interactivos, guías de usuario y soporte técnico en línea para garantizar que los usuarios comprendan y utilicen eficazmente todas las funcionalidades.

### 3.2.2. Requerimientos no funcionales

Los requerimientos no funcionales describen las cualidades del sistema, como su usabilidad, rendimiento, seguridad y compatibilidad, que aseguran que funcione de manera eficiente y segura.

ID	Requerimiento	Descripción
RNF001	Disponibilidad del Sistema	El sistema debe estar disponible el 99.9% del tiempo para garantizar acceso continuo a los usuarios y minimizar interrupciones en el servicio.
RNF002	Tiempo de Respuesta	Las operaciones críticas, como el acceso al dashboard o la generación de informes, deben completarse en menos de 2 segundos para optimizar la experiencia del usuario.
RNF003	Seguridad de la Información	Toda la información personal y de progreso debe estar protegida mediante cifrado en tránsito (SSL) y en reposo, asegurando la privacidad de los usuarios.
RNF004	Autenticación y Control de	El sistema debe implementar autenticación basada en

	Acceso	roles (e.g., administrador, usuario estándar) y restringir el acceso a funciones específicas según los permisos asignados.
RNF005	Escalabilidad	El sistema debe ser capaz de soportar un crecimiento en el número de usuarios y datos sin degradar el rendimiento, adaptándose a futuras expansiones de funcionalidad.
RNF006	Compatibilidad Multiplataforma	La plataforma debe ser completamente accesible desde dispositivos móviles, tabletas y computadoras con navegadores populares, como Chrome, Safari y Firefox.
RNF007	Usabilidad	La interfaz debe ser intuitiva y fácil de usar, asegurando que los usuarios puedan navegar y realizar tareas con un mínimo de aprendizaje.
RNF008	Documentación Técnica y de Usuario	El sistema debe incluir documentación completa, tanto técnica como de usuario, para facilitar el mantenimiento, el desarrollo futuro y el uso eficaz por parte de los usuarios finales.
RNF009	Respaldo Automático	La plataforma debe realizar copias de seguridad automáticas diarias de la base de datos para garantizar la recuperación de datos en caso de fallos.
RNF010	Recuperación ante Fallos	El sistema debe ser capaz de recuperarse de fallos críticos en menos de una hora para minimizar el impacto en los usuarios y operaciones.
RNF011	Interoperabilidad	El sistema debe poder integrarse fácilmente con otras plataformas o sistemas futuros que la universidad pueda implementar, como módulos de fisioterapia o bienestar mental.
RNF012	Accesibilidad Universal	El diseño del sistema debe cumplir con los estándares de accesibilidad, como WCAG 2.1, para asegurar que personas con discapacidades puedan utilizar la plataforma sin barreras.
RNF013	Mantenibilidad	El sistema debe ser modular, facilitando las actualizaciones y la incorporación de nuevas funcionalidades con un esfuerzo mínimo.
RNF014	Pruebas de Calidad	El sistema debe ser sometido regularmente a pruebas de funcionalidad, usabilidad y rendimiento para garantizar su operación óptima.
RNF015	Optimización del Consumo de Recursos	La plataforma debe estar diseñada para funcionar eficientemente en términos de uso de memoria y procesamiento, incluso en dispositivos de gama media o baja.

### **3.3. Identificación de Actores del Sistema**

La identificación de los actores del sistema es crucial para entender quiénes interactuarán con la plataforma **HealthyFit** y cuáles serán sus necesidades y responsabilidades. Cada uno de estos actores tienen roles específicos que deben ser atendidos para garantizar el funcionamiento eficiente, seguro y eficaz del sistema. A continuación, se describen los actores principales y sus roles dentro del sistema:

#### **3.3.1. Administrador del Sistema**

##### **Descripción:**

El Administrador del Sistema es el responsable de la configuración inicial, mantenimiento continuo y supervisión técnica de la plataforma. Este actor tiene acceso a todas las funcionalidades críticas para garantizar el funcionamiento óptimo y la seguridad del sistema.

##### **Funciones:**

- Configuración inicial de la plataforma, incluyendo la creación de perfiles y permisos.
- Gestión y monitoreo de usuarios registrados.
- Supervisión del rendimiento y disponibilidad de la plataforma.
- Implementación de copias de seguridad automáticas y restauraciones en caso de fallos.
- Coordinación de actualizaciones, mantenimiento y solución de problemas técnicos.

### **3.3.2. Usuario Final (Oficinista)**

#### **Descripción:**

El Usuario Final es el oficinista (estudiante, personal administrativo o docente) que interactúa directamente con el sistema para gestionar su bienestar físico y recibir recomendaciones personalizadas.

#### **Funciones:**

- Registro y personalización de su perfil con metas de salud y preferencias.
- Creación, seguimiento y ajuste de rutinas de ejercicio personalizadas. Consulta y actualización de su progreso físico a través del dashboard interactivo.
- Generación de informes y visualización de recomendaciones específicas.
- Recepción de notificaciones sobre eventos de bienestar y logros alcanzados.

### **3.3.3. Administrador de Contenido**

#### **Descripción:**

El Administrador de Contenido es el encargado de gestionar la información y recursos disponibles en la plataforma, incluyendo rutinas predefinidas, planes alimenticios y guías interactivas.

#### **Funciones:**

- Actualización y publicación de contenido relacionado con rutinas y planes de alimentación.
- Gestión de material educativo y tutoriales para los usuarios.
- Moderación de eventos y actividades promovidas en la plataforma.
- Personalización de las recomendaciones genéricas según las necesidades de la comunidad universitaria.

### **3.3.4. Técnico de Soporte**

#### **Descripción:**

El Técnico de Soporte brinda asistencia técnica a los usuarios y al administrador del sistema, asegurando que cualquier incidencia sea resuelta de manera eficiente.

#### **Funciones:**

- Resolución de problemas técnicos reportados por los usuarios.
- Asistencia en la configuración inicial y ajustes del sistema.
- Capacitación de los usuarios y del administrador en el uso de nuevas funcionalidades.

- Monitoreo de la infraestructura tecnológica para evitar caídas o problemas de rendimiento.

### **3.3.5. Auditor del Sistema**

#### **Descripción:**

El Auditor del Sistema es el encargado de verificar que la plataforma cumpla con los estándares normativos, de calidad y seguridad exigidos por la Universidad Continental y la legislación vigente.

#### **Funciones:**

- Revisión periódica del cumplimiento normativo del sistema.
- Auditorías de seguridad para garantizar la protección de datos personales y confidenciales.
- Generación de informes de calidad y recomendaciones de mejora para el sistema.
- Validación de la efectividad de las actualizaciones y mejoras implementadas.

## CAPÍTULO 4

### PLANIFICACIÓN DEL PROYECTO

#### 4.1. Definición de Roles de Trabajo

##### 4.1.1. Product owner

Nilton Nolasco Meza

##### 4.1.2. Scrum master

Jeanpier Taype Huaroc

##### 4.1.3. Team member

- Taype Huarocc Jeanpier
- Emil Inga Barzola
- Luis Salcedo
- William Ramirez
- Alexander Ramos
- Nilton Nolasco

##### 4.1.4. Tester

William Ramirez Crispin

##### 4.1.5 Diseñador UX/UI

Nilton Nolasco Meza

## 4.2. Product Backlog

ID Historia de Usuario	Como (Rol)	Deseo...	Para...	ID Tarea	Descripción de la Tarea
HU001	Usuario	Registrarme en la plataforma	Acceder a las funcionalidades del sistema.	1.1	Crear formulario de registro.
				1.2	Implementar validaciones de datos.
				1.3	Guardar los datos en la base de datos MongoDB.
HU002	Usuario	Iniciar sesión	Ver mi información personalizada.	2.1	Crear formulario de inicio de sesión.
				2.2	Implementar lógica para autenticación con email y contraseña.
HU003	Administrador	Definir roles de usuario y admin	Gestionar permisos y funcionalidades.	3.1	Configurar sistema de roles en MongoDB.
				3.2	Asignar permisos según el rol del usuario.
HU004	Administrador	Gestionar base de datos de usuarios	Controlar el acceso y roles.	4.1	Configurar conexión entre frontend y backend para gestionar usuarios.
				4.2	Validar integridad de datos en la base de datos.
HU005	Usuario	Tener roles vinculados a la base de datos	Asegurarme de que se asignen permisos correctamente.	5.1	Sincronizar los roles asignados con la base de datos MongoDB.
HU006	Administrador	Crear rutinas personalizadas	Ofrecer planes de ejercicios a los usuarios.	6.1	Diseñar la interfaz para la creación de rutinas.
				6.2	Crear lógica backend para almacenar rutinas en MongoDB.
				6.3	Validar rutinas creadas en el sistema.

HU007	Administrador	Asignar rutinas a usuarios	Que los usuarios reciban rutinas personalizadas.	7.1	Vincular rutinas personalizadas a perfiles de usuario.
				7.2	Notificar al usuario sobre la asignación de la rutina.
HU008	Usuario	Ver el dashboard con información actualizada	Monitorear mi progreso y estado.	8.1	Diseñar el dashboard con métricas clave del usuario.
				8.2	Implementar visualización de rutinas y progreso en tiempo real.
HU009	Administrador/Nutricionista	Crear dietas personalizadas	Proveer planes alimenticios específicos.	9.1	Diseñar interfaz para la creación de dietas personalizadas.
				9.2	Almacenar dietas creadas en MongoDB.
HU010	Administrador/Nutricionista	Asignar dietas a usuarios	Que los usuarios sigan planes nutricionales.	10.1	Vincular dietas creadas a perfiles de usuario.
				10.2	Notificar al usuario sobre la asignación de la dieta.
HU011	Usuario	Ver las dietas en el dashboard	Tener acceso a mis planes alimenticios.	11.1	Diseñar dashboard para visualizar dietas asignadas.

## 4.3. Sprint Backlog

### 4.3.1. Sprint 1

ID Tarea	Tarea	Responsable	ID Historia
1.1	Crear formulario de registro.	Nilton Nolasco, William Ramirez	HU001
1.2	Implementar validaciones de datos.	Luis Alcedo, Ramos Alexander	HU001
1.3	Guardar los datos en la base de datos MongoDB.	Emil Inga, Taype Huarocc	HU001
2.1	Crear formulario de inicio de sesión.	Nilton Nolasco, William Ramirez	HU002
2.2	Implementar lógica para autenticación con email y contraseña.	Tayne Huarocc, Emil Inga	HU002

#### 4.3.2. Sprint 2

ID Tarea	Tarea	Responsable	ID Historia
3.1	Configurar sistema de roles en MongoDB.	Emil Inga, Luis Alcedo	HU003
3.2	Asignar permisos según el rol del usuario.	Tayne Huarocc, Ramos Alexander	HU003
4.1	Configurar conexión entre frontend y backend para gestionar usuarios.	William Ramirez, Ramos Alexander	HU004
4.2	Validar integridad de datos en la base de datos.	Tayne Huarocc, Emil Inga	HU004

#### 4.3.3. Sprint 3

D Tarea	Tarea	Responsable	ID Historia
6.1	Diseñar la interfaz para la creación de rutinas.	William Ramirez, Nilton Nolasco	HU006
6.2	Crear lógica backend para almacenar rutinas en MongoDB.	Emil Inga, Luis Alcedo	HU006
6.3	Validar rutinas creadas en el sistema.	Tayne Huarocc, Ramos Alexander	HU006

#### 4.3.4. Sprint 4

D Tarea	Tarea	Responsable	ID Historia
9.1	Diseñar interfaz para la creación de dietas personalizadas.	William Ramirez, Nilton Nolasco	HU009
9.2	Almacenar dietas creadas en MongoDB.	Emil Inga, Luis Alcedo	HU009
10.1	Vincular dietas creadas a perfiles de usuario.	Tayne Huarocc, Ramos Alexander	HU010
10.2	Notificar al usuario sobre la asignación de la dieta.	Tayne Huarocc, William Ramirez	HU010

### 4.4. Planificación de Sprints

#### 4.4.1. Historias de usuario

HISTORIAS DE USUARIO	TAREAS
HU001	<ul style="list-style-type: none"> <li>- Crear formulario de registro de usuarios.</li> <li>- Implementar validaciones de datos en el registro.</li> </ul>
	<ul style="list-style-type: none"> <li>- Guardar los datos de usuarios en la base de datos MongoDB.</li> </ul>
HU002	<ul style="list-style-type: none"> <li>- Crear formulario de inicio de sesión.</li> <li>- Implementar lógica de autenticación con email y contraseña.</li> </ul>
HU003	<ul style="list-style-type: none"> <li>- Configurar sistema de roles en MongoDB.</li> <li>- Asignar permisos según el rol (Admin o Usuario).</li> </ul>
HU004	<ul style="list-style-type: none"> <li>- Configurar conexión entre frontend y backend para gestionar usuarios.</li> <li>- Validar integridad de datos en la base de datos MongoDB.</li> </ul>
HU005	<ul style="list-style-type: none"> <li>- Diseñar el dashboard principal para mostrar rutinas asignadas.</li> <li>- Implementar visualización del progreso del usuario en tiempo real.</li> </ul>
HU006	<ul style="list-style-type: none"> <li>- Diseñar la interfaz para la creación de rutinas personalizadas.</li> <li>- Crear lógica backend para almacenar rutinas personalizadas en MongoDB.</li> <li>- Validar rutinas creadas en el sistema.</li> </ul>
HU007	<ul style="list-style-type: none"> <li>- Diseñar interfaz para la creación de dietas personalizadas.</li> <li>- Vincular dietas creadas a perfiles de usuario.</li> <li>- Implementar notificaciones sobre asignación de dietas.</li> </ul>
HU008	<ul style="list-style-type: none"> <li>- Diseñar sección de visualización de rutinas y dietas en el dashboard del usuario.</li> <li>- Asegurar que las rutinas y dietas asignadas se actualicen correctamente en el sistema según el progreso del usuario.</li> </ul>

#### 4.4.2. Priorización de historias de usuario

ID Historia	Tarea	Prioridad
HU001	Crear formulario de registro de usuarios.	Alta
	Implementar validaciones de datos en el registro.	Alta
	Guardar los datos de usuarios en la base de datos MongoDB.	Media
HU002	Crear formulario de inicio de sesión.	Alta
	Implementar lógica de autenticación con email y contraseña.	Media
Sprint 1 (3 semanas)		
HU003	Configurar sistema de roles en MongoDB.	Alta
	Asignar permisos según el rol (Admin o Usuario).	Media
HU004	Configurar conexión entre frontend y backend para gestionar usuarios.	Media
	Validar integridad de datos en la base de datos MongoDB.	Media
Sprint 2 (3 semanas)		
HU005	Diseñar el dashboard principal para mostrar rutinas asignadas.	Alta
	Implementar visualización del progreso del usuario en tiempo real.	Alta
HU006	Diseñar la interfaz para la creación de rutinas personalizadas.	Alta

	Crear lógica backend para almacenar rutinas personalizadas en MongoDB.	Media
	Validar rutinas creadas en el sistema.	Media
<b>Sprint 3 (3 semanas)</b>		
<b>HU007</b>	Diseñar interfaz para la creación de dietas personalizadas.	Alta
	Vincular dietas creadas a perfiles de usuario.	Media
	Implementar notificaciones sobre asignación de dietas.	Media
<b>HU008</b>	Diseñar sección de visualización de rutinas y dietas en el dashboard del usuario.	Alta
	Asegurar que las rutinas y dietas asignadas se actualicen correctamente en el sistema.	Alta

#### 4.5. Cronograma de Actividades

NRO	Sprint	Actividad	Fecha de Inicio	Fecha de Fin	Duración (días)
1	SPRINT 0 (1 semana)	Gestión de proyecto	21/08/2024	27/08/2024	7
2	SPRINT 0 (1 semana)	Acuerdos de equipo	28/08/2024	3/09/2024	7
3	SPRINT 0 (1 semana)	Creación del acta de constitución	4/09/2024	10/09/2024	7
4	SPRINT 0 (1 semana)	Análisis de requerimientos y diseño	11/09/2024	17/09/2024	7

5	SPRINT 1 (3 semanas)	Crear formulario de registro de usuarios (HU001)	18/09/2024	22/09/2024	5
6	SPRINT 1 (3 semanas)	Implementar validaciones en el registro (HU001)	23/09/2024	27/09/2024	5
7	SPRINT 1 (3 semanas)	Guardar datos de usuarios en MongoDB (HU001)	28/09/2024	2/10/2024	5
8	SPRINT 1 (3 semanas)	Crear formulario de inicio de sesión (HU002)	3/10/2024	7/10/2024	5
9	SPRINT 1 (3 semanas)	Implementar autenticación (HU002)	8/10/2024	12/10/2024	5
10	SPRINT 2 (3 semanas)	Configurar sistema de roles en MongoDB (HU003)	13/10/2024	17/10/2024	5
11	SPRINT 2 (3 semanas)	Asignar permisos por rol (HU003)	18/10/2024	22/10/2024	5
12	SPRINT 2 (3 semanas)	Configurar conexión frontend-backend (HU004)	23/10/2024	27/10/2024	5
13	SPRINT 2 (3 semanas)	Validar integridad de datos (HU004)	28/10/2024	1/11/2024	5
14	SPRINT 3 (3 semanas)	Diseñar la interfaz para rutinas (HU006)	2/11/2024	6/11/2024	5
15	SPRINT 3 (3 semanas)	Crear lógica backend para rutinas (HU006)	7/11/2024	11/11/2024	5
16	SPRINT 3 (3 semanas)	Validar rutinas creadas (HU006)	12/11/2024	16/11/2024	5

17	SPRINT 3 (3 semanas)	Diseñar dashboard para usuarios (HU005)	17/11/2024	21/11/2024	5
18	SPRINT 4 (3 semanas)	Crear interfaz para dietas personalizadas (HU007)	22/11/2024	26/11/2024	5
19	SPRINT 4 (3 semanas)	Vincular dietas a usuarios (HU007)	27/11/2024	1/12/2024	5
20	SPRINT 4 (3 semanas)	Notificar asignación de dietas (HU007)	2/12/2024	3/12/2024	2

#### 4.6. Gestión de Riesgos

Nombre del Proyecto: HealthyFit																																																																																											
<p><b>Umbral para Amenazas</b></p> <table border="1"> <tr> <td>Alto</td> <td>4,5</td> </tr> <tr> <td>Medio</td> <td>2</td> </tr> </table> <p><b>Umbral para Oportunidades</b></p> <table border="1"> <tr> <td>Alto</td> <td>4</td> </tr> <tr> <td>Medio</td> <td>2</td> </tr> </table>										Alto	4,5	Medio	2	Alto	4	Medio	2																																																																										
Alto	4,5																																																																																										
Medio	2																																																																																										
Alto	4																																																																																										
Medio	2																																																																																										
<p><b>Instrucciones:</b></p> <ol style="list-style-type: none"> <li>1. Ingrese la puntuación de Impacto y Probabilidad (o use la que viene por defecto)</li> <li>2. Ingrese el Umbral para Amenazas (alto y medio)</li> <li>3. Ingrese el Umbral para Oportunidades (alto y medio)</li> <li>4. Ingrese los riesgos y oportunidades son sus respectivos impactos y probabilidades en las hojas respectivas. La codificación de color le ayudará a visualizar fácilmente.</li> </ol>																																																																																											
<table border="1"> <thead> <tr> <th colspan="5">Amenazas</th> <th colspan="5">Oportunidades</th> </tr> </thead> <tbody> <tr> <td></td> <td><span style="background-color: red;"> </span></td> <td><span style="background-color: yellow;"> </span></td> <td><span style="background-color: green;"> </span></td> <td></td> <td><span style="background-color: blue;"> </span></td> <td><span style="background-color: yellow;"> </span></td> <td><span style="background-color: green;"> </span></td> <td></td> <td></td> </tr> <tr> <td></td> <td>= Alta Prioridad</td> <td>= Mediana Prioridad</td> <td>= Baja Prioridad</td> <td></td> <td>= Alta Prioridad</td> <td>= Mediana Prioridad</td> <td>= Baja Prioridad</td> <td></td> <td></td> </tr> </tbody> </table>										Amenazas					Oportunidades						<span style="background-color: red;"> </span>	<span style="background-color: yellow;"> </span>	<span style="background-color: green;"> </span>		<span style="background-color: blue;"> </span>	<span style="background-color: yellow;"> </span>	<span style="background-color: green;"> </span>				= Alta Prioridad	= Mediana Prioridad	= Baja Prioridad		= Alta Prioridad	= Mediana Prioridad	= Baja Prioridad																																																						
Amenazas					Oportunidades																																																																																						
	<span style="background-color: red;"> </span>	<span style="background-color: yellow;"> </span>	<span style="background-color: green;"> </span>		<span style="background-color: blue;"> </span>	<span style="background-color: yellow;"> </span>	<span style="background-color: green;"> </span>																																																																																				
	= Alta Prioridad	= Mediana Prioridad	= Baja Prioridad		= Alta Prioridad	= Mediana Prioridad	= Baja Prioridad																																																																																				
<table border="1"> <thead> <tr> <th colspan="6">Amenazas</th> <th colspan="4">Oportunidades</th> </tr> <tr> <th rowspan="6">Impacto</th> <th>9</th> <th>0,9</th> <th>2,7</th> <th>4,5</th> <th>6,3</th> <th>8,1</th> <th>8,1</th> <th>6,3</th> <th>4,5</th> <th>2,7</th> <th>0,9</th> </tr> </thead> <tbody> <tr> <td>7</td> <td>0,7</td> <td>2,1</td> <td>3,5</td> <td>4,9</td> <td>6,3</td> <td>6,3</td> <td>6,3</td> <td>4,9</td> <td>3,5</td> <td>2,1</td> <td>0,7</td> </tr> <tr> <td>5</td> <td>0,5</td> <td>1,5</td> <td>2,5</td> <td>3,5</td> <td>4,5</td> <td>4,5</td> <td>4,5</td> <td>3,5</td> <td>2,5</td> <td>1,5</td> <td>0,5</td> </tr> <tr> <td>3</td> <td>0,3</td> <td>0,9</td> <td>1,5</td> <td>2,1</td> <td>2,7</td> <td>2,7</td> <td>2,7</td> <td>2,1</td> <td>1,5</td> <td>0,9</td> <td>0,3</td> </tr> <tr> <td>1</td> <td>0,1</td> <td>0,3</td> <td>0,5</td> <td>0,7</td> <td>0,9</td> <td>0,9</td> <td>0,9</td> <td>0,7</td> <td>0,5</td> <td>0,3</td> <td>0,1</td> </tr> <tr> <td></td> <td>0,1</td> <td>0,3</td> <td>0,5</td> <td>0,7</td> <td>0,9</td> <td></td> <td>0,9</td> <td>0,7</td> <td>0,5</td> <td>0,3</td> <td>0,1</td> </tr> </tbody> </table>										Amenazas						Oportunidades				Impacto	9	0,9	2,7	4,5	6,3	8,1	8,1	6,3	4,5	2,7	0,9	7	0,7	2,1	3,5	4,9	6,3	6,3	6,3	4,9	3,5	2,1	0,7	5	0,5	1,5	2,5	3,5	4,5	4,5	4,5	3,5	2,5	1,5	0,5	3	0,3	0,9	1,5	2,1	2,7	2,7	2,7	2,1	1,5	0,9	0,3	1	0,1	0,3	0,5	0,7	0,9	0,9	0,9	0,7	0,5	0,3	0,1		0,1	0,3	0,5	0,7	0,9		0,9	0,7	0,5	0,3	0,1
Amenazas						Oportunidades																																																																																					
Impacto	9	0,9	2,7	4,5	6,3	8,1	8,1	6,3	4,5	2,7	0,9																																																																																
	7	0,7	2,1	3,5	4,9	6,3	6,3	6,3	4,9	3,5	2,1	0,7																																																																															
	5	0,5	1,5	2,5	3,5	4,5	4,5	4,5	3,5	2,5	1,5	0,5																																																																															
	3	0,3	0,9	1,5	2,1	2,7	2,7	2,7	2,1	1,5	0,9	0,3																																																																															
	1	0,1	0,3	0,5	0,7	0,9	0,9	0,9	0,7	0,5	0,3	0,1																																																																															
		0,1	0,3	0,5	0,7	0,9		0,9	0,7	0,5	0,3	0,1																																																																															
<p><b>Probabilidad</b></p>																																																																																											

Registro de Riesgos							
Nombre del Proyecto: [REDACTED]							
ID Riesgo	Descripción del Riesgo		Área de Impacto	Causa	Impacto	Probabilidad	Puntuación de Riesgo
1.1	Nuevas tecnologías de Desarrollo web de uso de Full stack Mern	Cronograma	El equipo no tiene experiencia en dicha tecnología	Muy Serio	Alto	Alto	4,9
1.2	Confusión y entregas apresuradas al no verificar la actualización de documentos	Cronograma	Falta de comunicación entre grupos	Serio	Medio	Medio	2,5
1.3	Las estimaciones de tiempo y esfuerzo demasiado optimistas pueden generar plazos poco realistas y retrasos	Cronograma	Falta de evaluación cuidadosa	Muy Serio	Alto	Alto	4,9
1.4	Errores de codificación y problemas técnicos que afectan la funcionalidad del aplicativo	Calidad	El equipo no tiene experiencia en dicha tecnología	Criticó	Alto	Alto	6,3
1.5	Cambios frecuentes en los requisitos del aplicativo durante el desarrollo	Calidad	El equipo no define correctamente los requisitos	Criticó	Medio	Medio	4,5
1.6	Acceso no autorizado a datos confidenciales, como información de pacientes	Calidad	Falta de medidas de seguridad adecuadas	Criticó	Alto	Alto	6,3
1.7	Subestimación de los costos involucrados en el desarrollo del aplicativo	Costo	Evaluación insuficiente de los requisitos del proyecto, falta de experiencia en la estimación de costos	Muy Serio	Medio	Medio	3,5
1.8	Retrasos en el desarrollo debido a problemas técnicos, cambios en los requisitos o falta de recursos	Costo	Problemas técnicos inesperados, falta de recursos o habilidades específicas	Muy Serio	Alto	Alto	4,9
1.9	Expansión no controlada de las características y funcionalidades del aplicativo durante el desarrollo	Alcance	Presiones del cliente o stakeholders para agregar nuevas funcionalidades	Serio	Medio	Medio	2,5
	Configuraciones de seguridad, los permisos o la configuración del servidor	Calidad		Criticó	Medio	Medio	4,5

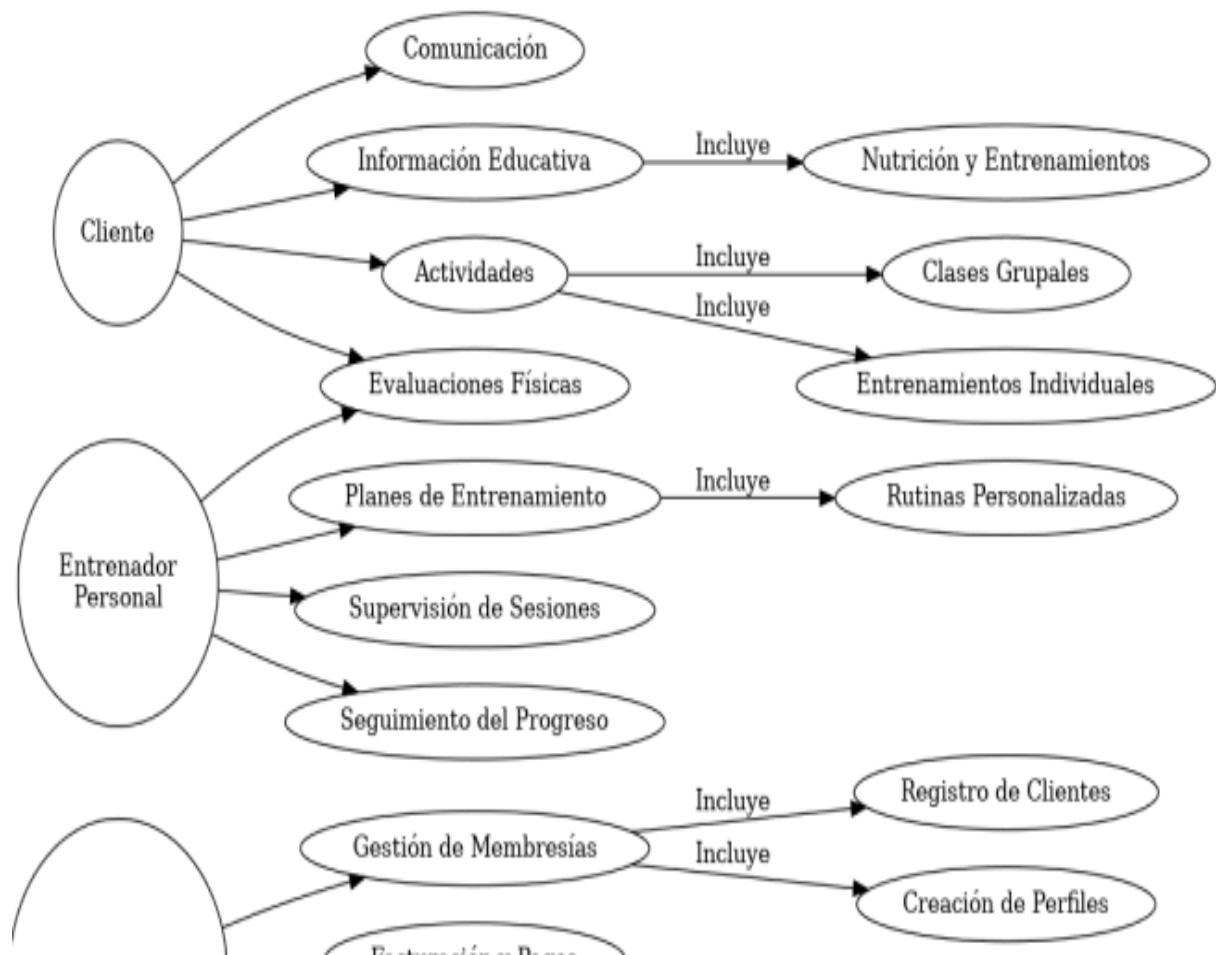
Registro de Oportunidades							
Nombre del Proyecto: [REDACTED]							
ID Oportunidad	Descripción de la Oportunidad		Área de Impacto	Causa de la Oportunidad	Impacto	Probabilidad	Puntuación de la Oportunidad
O1.1	Implementación exitosa de nuevas tecnologías con mayor eficiencia en el desarrollo	Calidad	Tecnología avanzada adoptada rápidamente	Alto	Alto	Alto	4,9
O1.2	Definición clara de requisitos desde el inicio del proyecto que reduce el re-trabajo	Alcance	Colaboración efectiva con stakeholders	Alto	Medio	Medio	3,5
O1.3	Optimización de los costos al identificar tecnologías más económicas sin perder calidad	Costo	Evaluación efectiva de opciones tecnológicas	Alto	Alto	Alto	4,9
O1.4	Mejora en la seguridad del sistema al implementar protocolos avanzados	Calidad	Implementación de nuevas políticas de seguridad	Muy Alto	Medio	Medio	4,5
O1.5	Uso de metodologías ágiles que mejoran la adaptabilidad del equipo a cambios	Cronograma	Implementación efectiva de Scrum	Alto	Alto	Alto	4,9
O1.6	Capacitación continua del equipo que reduce errores y mejora la calidad final del producto	Calidad	Formación regular y actualización de conocimientos	Alto	Alto	Alto	4,9
O1.7	Uso de pruebas automatizadas que reduce tiempo de desarrollo y mejora la calidad	Calidad	Implementación de herramientas de automatización	Muy Alto	Alto	Alto	6,3
O1.8	Obtención de retroalimentación temprana del cliente que mejora la satisfacción y la calidad del producto	Calidad	Interacciones frecuentes con el cliente	Alto	Medio	Medio	3,5
O1.9	Identificación temprana de riesgos que permite una gestión más efectiva del proyecto	Cronograma	Proceso de evaluación de riesgos continuo	Alto	Alto	Alto	4,9
O2.0	Flexibilidad en el diseño del sistema que permite adaptarse fácilmente a nuevas demandas del mercado	Alcance	Arquitectura del sistema diseñada modularmente	Alto	Medio	Medio	3,5
O2.1	Comunicación efectiva entre los equipos que mejora la coordinación y el rendimiento general del proyecto	Cronograma	Uso de herramientas colaborativas y reuniones regulares	Alto	Alto	Alto	4,9

## CAPÍTULO 5

### DISEÑO DEL SISTEMA DE INFORMACIÓN

#### 5.1. Diseño de Diagramas UML

##### 5.1.1. Diagramas de casos de uso



### 5.1.2. Diagramas de flujo de datos

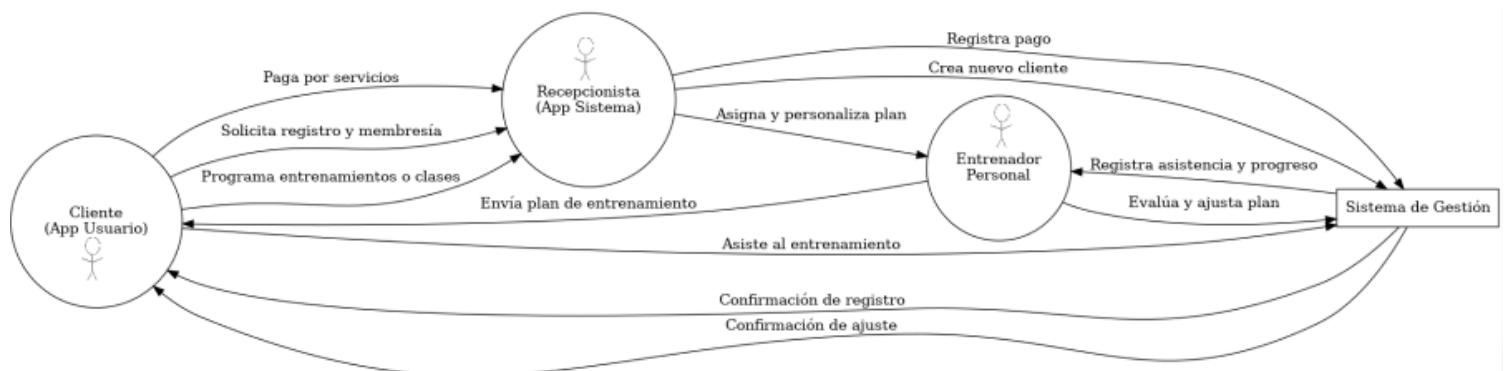
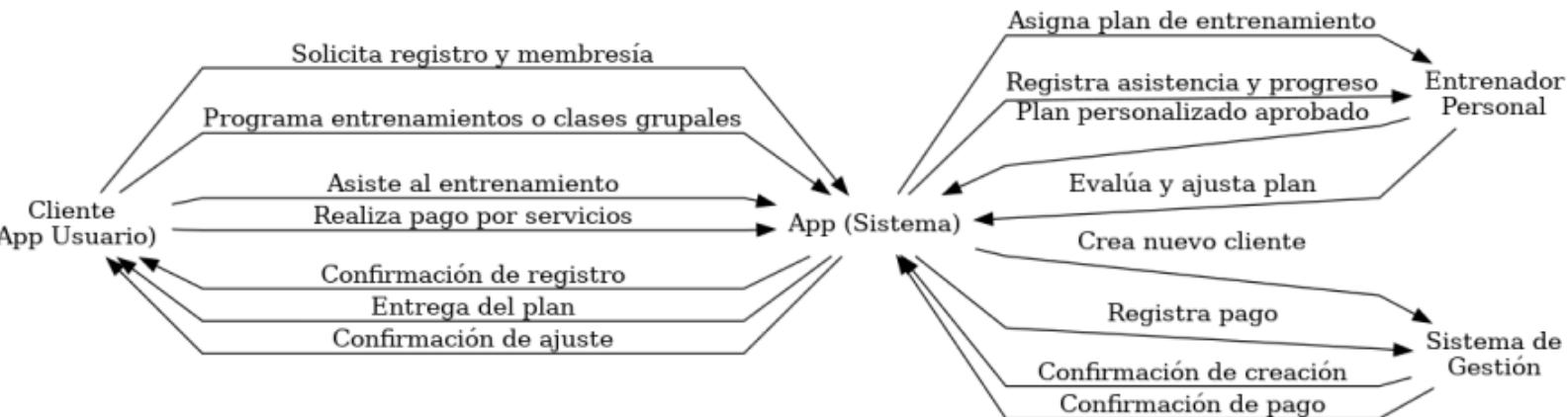
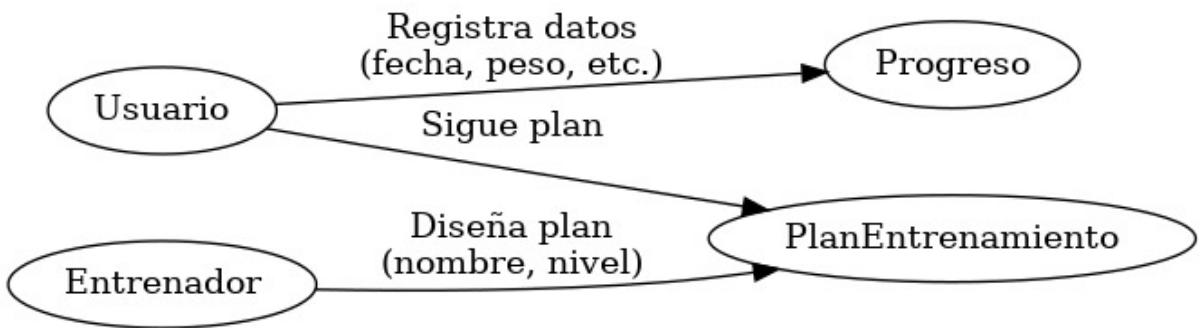


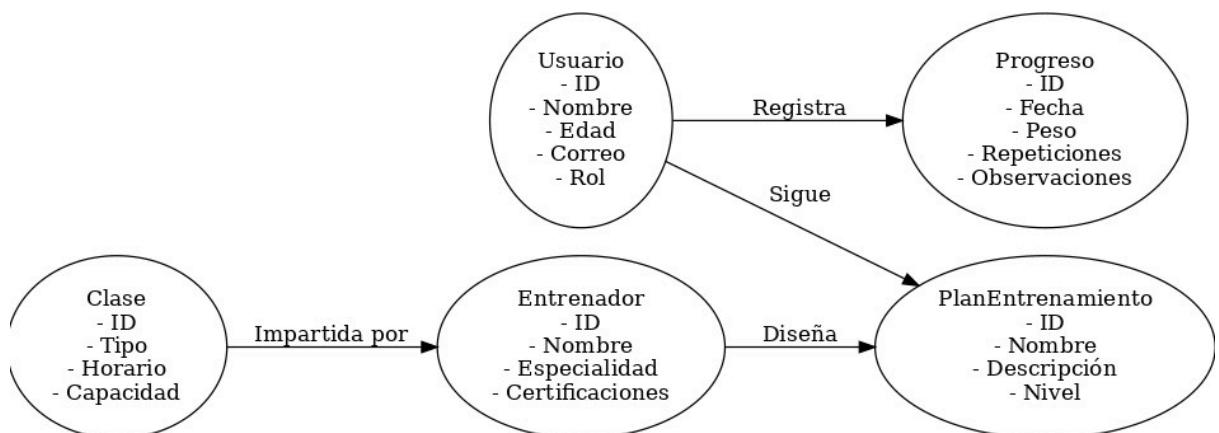
diagrama de contexto



### 5.1.3. Diagramas de casos de Uso

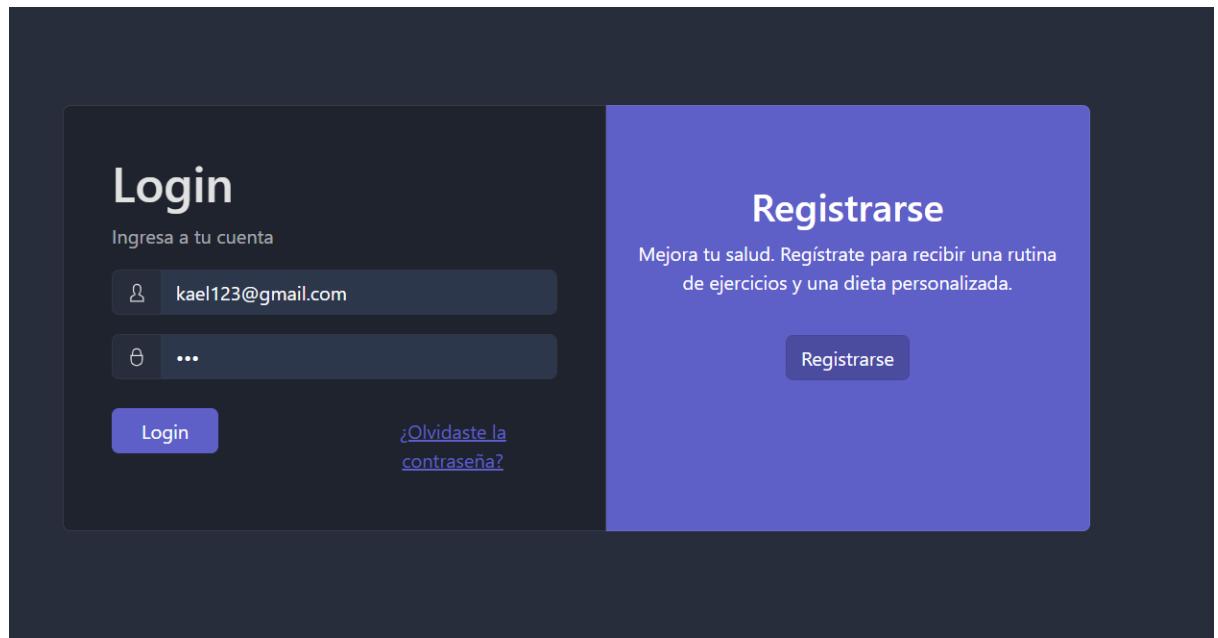
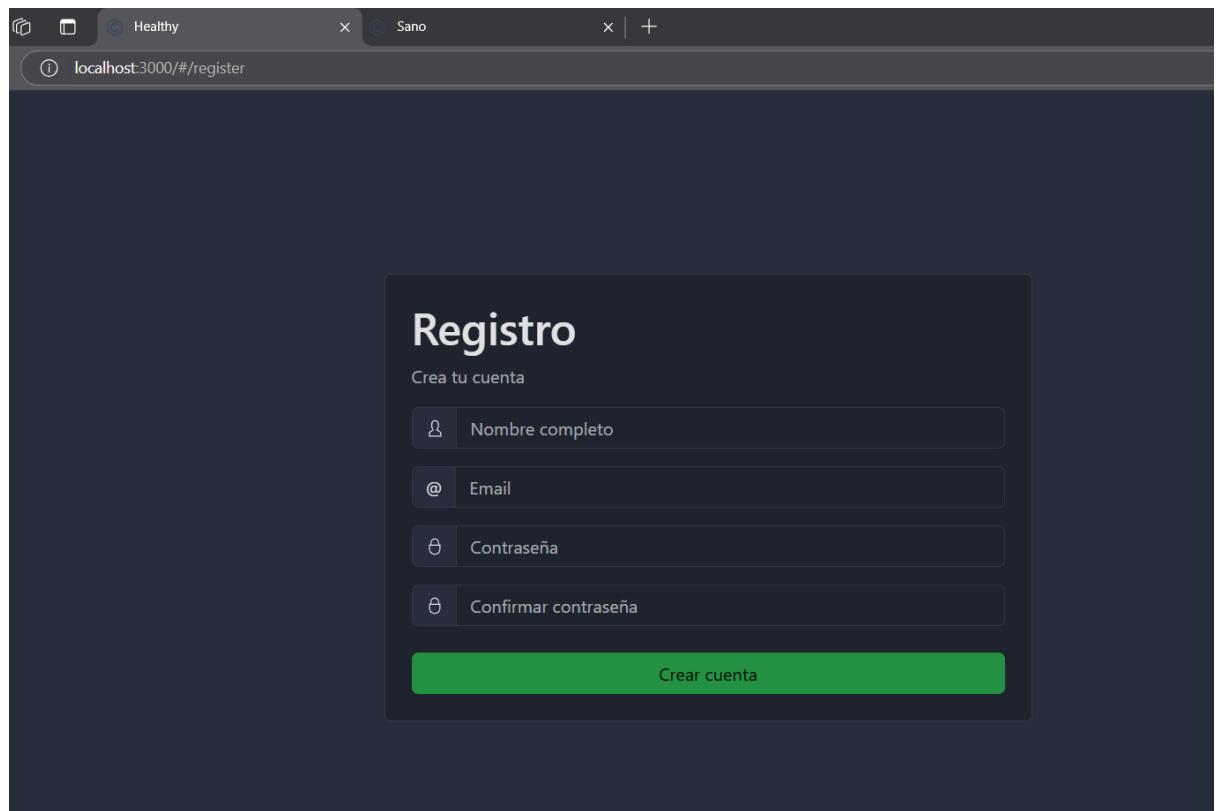


### 5.1.4. Diagramas de caso de Uso



## 5.2. Diseño de Interfaces Básicas

### 5.2.1. Acceso login



## 5.2.2. Interfaz ...

The screenshot displays a dark-themed user interface for a fitness application. On the left, a sidebar contains navigation links: Dashboard, Dietas, Rutinas ejercicios (which is selected and highlighted in blue), Dashboard Admin, and Suscribirse. The main content area has a header "Rutina / ejercicios" and a sub-header "ABDOMINALES SERIES XXXXXXXXX". A blue button labeled "Ver Demostración" is visible. Below this, a video player shows a video titled "Ejercicios de abdominales contra la pared I MARCA" from the channel "DEPORTE IH". The video player includes controls like play/pause, volume, and a progress bar showing 0:00 / 0:48. To the right of the video, there's a "Compartir" (Share) button and other video thumbnails for related content such as "Ejercicios en Pared", "Glúteos Abdomen Piernas", "EXERCISE 1", "EXERCISE 2", "15 MIN FAT BURNING WALL PILATES", "HIPOPRESIVOS", "¿Cómo hacerlos?", and "ADIOS BARRIGA". The bottom right corner of the video player shows "YouTube".

#	Nombre	Email	Acciones
1	73825204@continental.edu.pe	lucash123@gmail.com	Editar Eliminar Asignar Rutina Asignar Dieta
2	lucas	kael123@gmail.com	Editar Eliminar Asignar Rutina Asignar Dieta

## CAPÍTULO 6

### CODIFICACIÓN DEL SOFTWARE

#### 6.1 Desarrollo del Sprint 1

##### 6.1.1 Sprint planning

###### Objetivo del Sprint 1:

- Desarrollar las funcionalidades básicas de registro e inicio de sesión para los usuarios.
- Asegurar que el sistema de autenticación esté implementado correctamente para continuar con el desarrollo.

###### Actividades planificadas:

- Crear formulario de registro para nuevos usuarios.
- Implementar sistema de inicio de sesión (correo y contraseña).
- Establecer los roles de Admin y Usuario en el sistema.
- Desarrollar validaciones para asegurar la correcta autenticación y registro.

### 6.1.2 Sprint backlog



Fig. Sprint backlog - sprint 1

### 6.1.3 Historias de usuarios

ID Historia de Usuario	Como	Deseo	Para
HU001	Usuario	Registrarme en la plataforma	Acceder a las funcionalidades del sistema
HU002	Usuario	Iniciar sesión en mi cuenta	Acceder a mi información SPRINpersonalizada
HU003	Admin	Definir los roles de usuario y admin	Asignar permisos y funcionalidades a cada tipo de usuario

Tabla. Historia de Usuarios Sprint 1

### 6.1.4 Taskboard

ID Tarea	Descripcion	ID Historia
T1.1	Implementar formulario de registro	1
T1.2	Validar datos de registro	1
T1.3	Implementar inicio de sesión	2
T1.4	Pruebas de seguridad para login/registro	2
T1.5	Validar rol (Usuario/Admin)	3

Fig. Taskboard - Sprint 1

### **6.1.5 Daily scrum**

#### **MINUTA DE REUNIÓN #1**

**LUGAR:** Google Meet

**FECHA:** 24 de agosto de 2024

**HORA:** 8:00 pm

**OBJETIVO:** Evaluar avances en el registro de usuarios e inicio de sesión.

#### **Participantes:**

- Alcedo Muñoz Luis Fernando
- Navarro Arias Michael Brandon
- Nolasco Meza Nilton Anthony
- Paucarchuco Ramos Alexander David
- Ramirez Crispin William Alexis

#### **Resumen de la reunión:**

- El equipo avanzó en la creación del formulario de registro.
- La funcionalidad de inicio de sesión se está desarrollando correctamente con autenticación por correo y contraseña.
- Aún faltan algunas pruebas de validación de datos en el registro e inicio de sesión.

#### **Acuerdos:**

- Continuar con el desarrollo del inicio de sesión con autenticación en 2 pasos (correo y contraseña).
- Establecer roles por defecto para Admin y Usuario.

### 6.1.6 Sprint review

#### Historias de Usuario completadas en Sprint 1:

- **HU001:** Registro de usuarios.
- **HU002:** Inicio de sesión de usuarios.
- **HU003:** Creación de roles Admin y Usuario.

#### Métricas clave:

- Historias de usuario completadas: 3/3 (100%)
- Retraso acumulado: 0 días

#### Próximos avances:

- Empezar a trabajar en la creación de rutinas personalizadas y dietas para el Admin en el siguiente sprint.

### 6.1.7 Criterios de aceptación

ID HU	Descripción de la Tarea	Criterio de aceptación
HU001	Desarrollar formulario de registro de usuario	El formulario debe capturar nombre, correo electrónico, y contraseña. Los campos son obligatorios.
HU002	Crear funcionalidad de inicio de sesión	El sistema debe permitir el inicio de sesión con correo y contraseña válidos.
HU003	Crear roles de Admin y Usuario con permisos básicos	Los usuarios deben ser asignados automáticamente al rol de "Usuario". El Admin debe tener privilegios para crear, modificar y eliminar contenido.

Tabla. Tabla de criterios de aceptación - Sprint 1

## 6.1.8 Resultados del sprint

### 6.1.8.1 Evidencias.

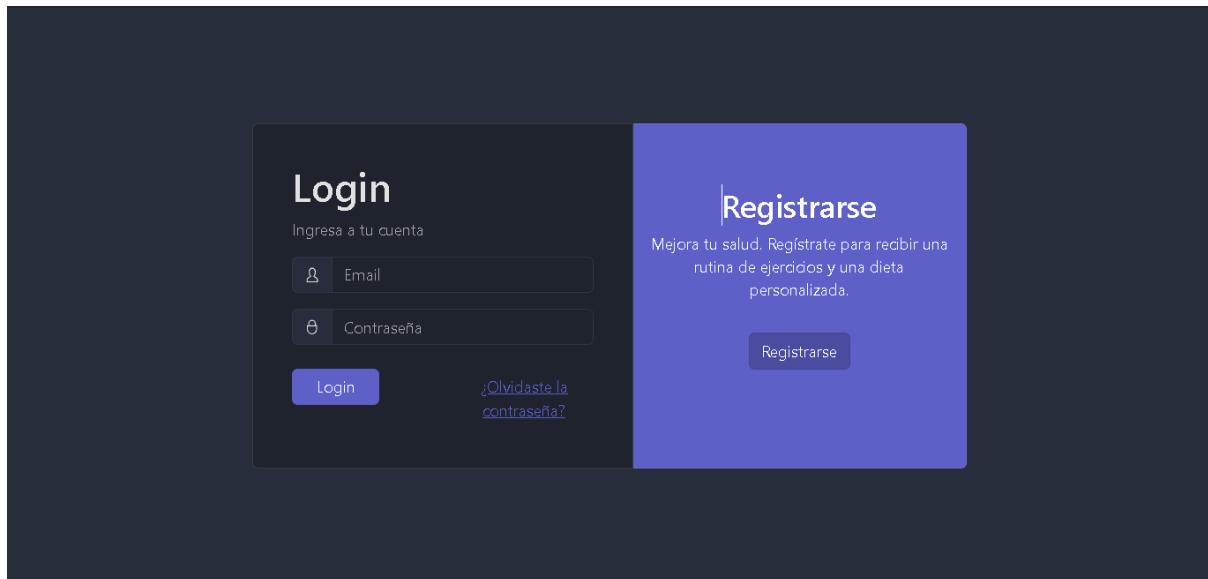


Fig. Login implementado

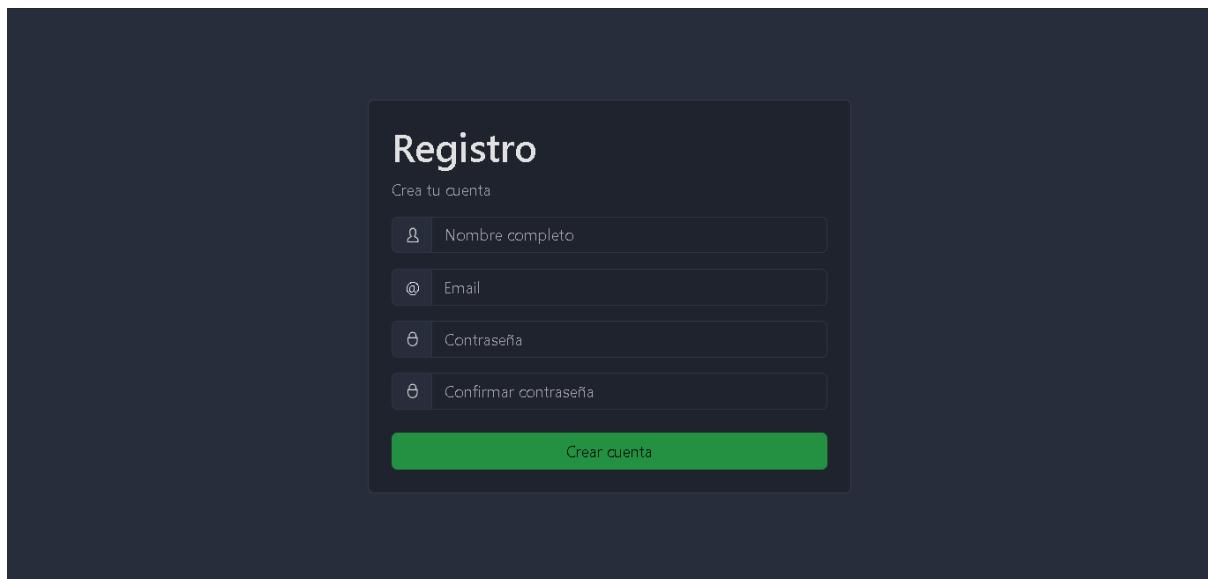


Fig. Registro implementado

### 6.1.8.2 Prueba de desarrollo.

The screenshot shows the Cypress Test Runner interface. On the left, the 'Golden\_path\_login.cy.js' file is open, displaying a test script for logging in. The script uses the 'viimport' command to open the browser, visits the login page, and fills in email and password fields before clicking the login button. A note at the bottom indicates a POST request to the API. On the right, the browser window shows the 'Login' page with fields for email and password, and a 'Login' button. To the right of the login form is a sidebar with a 'Registrarse' button and a promotional message about improving health through exercise routines and personalized diets.

Fig. Prueba Golden path Login

The screenshot shows the Cypress Test Runner interface. On the left, the 'Happy\_path\_register.cy.js' file is open, displaying a test script for registering a new user. The script fills in first name, last name, email, and password fields, then clicks the registration button. A note at the bottom indicates a POST request to the API. On the right, the browser window shows the 'Registro' (Registration) page with fields for first name, last name, email, and password, and a 'Crear cuenta' (Create account) button. A success message at the top of the page reads 'Registration successful! Please log in.'

Fig. Prueba Golden path Registro

### 6.1.9 Sprint retrospective

#### ¿Qué aprendimos durante el Sprint 1?

- La importancia de tener un flujo claro y simple para el registro e inicio de sesión de los usuarios.

- La asignación de roles es crucial para gestionar el acceso a las funcionalidades del sistema.

### **¿Qué hicimos bien?**

- La implementación de roles y autenticación fue exitosa.
- La comunicación entre los miembros del equipo fue fluida y eficiente.

### **¿Qué podemos mejorar?**

- Mejorar la interfaz de usuario en el formulario de registro e inicio de sesión.
- Realizar más pruebas de integración para asegurar el funcionamiento correcto del sistema.

### **Acciones a realizar para el siguiente sprint:**

- Empezar el desarrollo de las funcionalidades de rutinas personalizadas y dietas para el admin.
- Revisar las validaciones de roles para asegurarnos de que los administradores puedan gestionar los usuarios correctamente.

## **6.2 Desarrollo del Sprint 2**

### **6.2.1 Sprint planning**

#### **Objetivo del Sprint 2:**

- Implementar el sistema de registro e inicio de sesión en la web.
- Mejorar la base de datos en MongoDB, vinculando los roles de usuario y admin con la web.

### Actividades planificadas:

- Completar la base de datos en MongoDB.
- Integrar MongoDB con el sistema de registro y roles.
- Asegurar que los datos se gestionen correctamente en la plataforma.

### 6.2.2 Sprint backlog

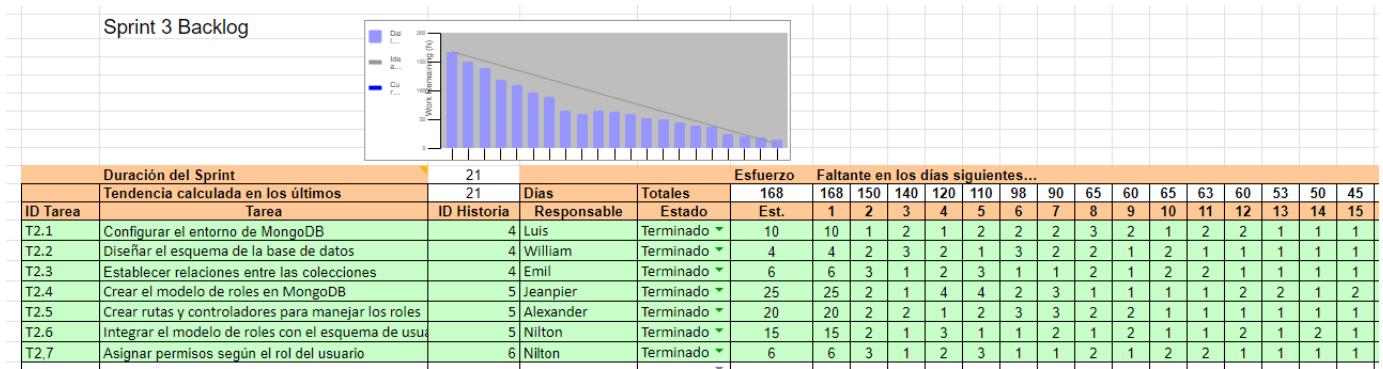


Fig. Sprint backlog - Sprint 2

### 6.2.3 Historias de usuarios

ID Historia de Usuario	Como	Deseo	Para
HU001	Admin	Gestionar base de datos de usuarios	Controlar el acceso y roles de los usuarios en la plataforma
HU002	Usuario	Vincular roles con la base de datos	Asegurarse de que se me asignen los permisos correctos
HU003	Admin	Validar roles con base de datos	Asegurar que los usuarios y administradores puedan realizar tareas correctamente

Tabla. Historia de Usuarios Sprint 2

#### 6.2.4 Taskboard

ID Tarea	Tarea	ID Historia
T2.1	Configurar el entorno de MongoDB	4
T2.2	Diseñar el esquema de la base de datos	4
T2.3	Establecer relaciones entre las colecciones	4
T2.4	Crear el modelo de roles en MongoDB	5
T2.5	Crear rutas y controladores para manejar los roles	5
T2.6	Integrar el modelo de roles con el esquema de usuario	5
T2.7	Asignar permisos según el rol del usuario	6

Fig. Taskboard - Sprint 1

#### 6.2.5 Daily scrum

##### MINUTA DE REUNIÓN #2

**LUGAR:** Google Meet

**FECHA:** 1 de septiembre de 2024

**HORA:** 8:00 pm

**OBJETIVO:** Revisión de la integración de la base de datos con MongoDB.

##### Participantes:

- Alcedo Muñoz Luis Fernando
- Navarro Arias Michael Brandon
- Nolasco Meza Nilton Anthony
- Paucarchuco Ramos Alexander David
- Ramirez Crispin William Alexis

##### Resumen de la reunión:

- La base de datos MongoDB se configuró correctamente.
- La integración con los roles se realizó con éxito, permitiendo gestionar los permisos.

#### **Acuerdos:**

- Validar todos los roles en la base de datos.
- Probar la integración en el entorno de producción.

#### **6.2.6 Sprint review**

##### **Historias de Usuario completadas en Sprint 2:**

- **HU004:** Implementación de base de datos MongoDB.
- **HU005:** Integración de roles con MongoDB.
- **HU006:** Validación de roles con base de datos.

##### **Métricas clave:**

- Historias de usuario completadas: 3/3 (100%)
- Retraso acumulado: 0 días

#### **6.2.7 Criterios de aceptación**

ID HU	Descripción de la Tarea	Descripción de la Tarea
HU004	Implementación de base de datos MongoDB	La base de datos MongoDB debe estar configurada correctamente y ser accesible desde la aplicación, permitiendo la gestión eficiente de los datos de los usuarios.
HU005	Integración de roles con MongoDB	El sistema debe permitir asignar un rol (Admin o Usuario) a cada usuario y almacenar esa información correctamente en la

		base de datos MongoDB.
HU006	Validación de roles con base de datos	El sistema debe validar que el rol del usuario en la base de datos coincida con los permisos asignados al usuario, garantizando un acceso adecuado a las funcionalidades según el rol.

Tabla. Tabla de criterios de aceptación - Sprint 2

## 6.2.8 Resultados del sprint 2

### 6.2.8.1 Evidencias.

The screenshot shows the MongoDB Compass interface connected to the 'localhost:27017' database and the 'db\_node' collection. The 'users' collection contains three documents:

- Entrenador**: name: "Entrenador", email: "entrenador@gmail.com", password: "\$2a\$10\$2PanfSG92Mz6PZyG8gTbtOUF1PaVm3fEXv7FNSGNZ99H10bogAK\*", role: "admin", \_\_v: 0
- marco**: name: "marco", email: "marco@gmail.com", password: "\$2a\$10\$J3QNPsD9W136Dj9zYJZl6ew4g/alkPuQgqndavxRn6RFc9vdwMP-i\*", role: "user", \_\_v: 0
- William**: name: "William", email: "william@gmail.com", password: "\$2a\$10\$gYqhW3GyFjXFSks114bhHux/Dp.lTNE3hLD16CPc1S87vp4nsGV2W\*", role: "admin", \_\_v: 0

Fig. Base de datos

The screenshot shows the MongoDB Compass application interface. The left sidebar displays connections to 'localhost:27017' and its databases: 'admin', 'config', 'db\_node', 'local', and 'notesdb'. The 'db\_node' database is selected, and its 'diets' collection is shown in the main pane. The 'Documents' tab is active, displaying four document entries. Each entry includes fields such as '\_id', 'user', 'foodName', 'calories', 'description', and '\_v'. The first document's description is 'Comida A, Comida B, Comida C, Comida D'. The second's is 'Alimento A, '. The third's is 'desadssda1l321'. The fourth's is 'desadasdasdasdasdasda'. The bottom right of the interface shows a toolbar with buttons for 'ADD DATA', 'EXPORT DATA', 'UPDATE', 'DELETE', 'Explain', 'Reset', 'Find', and 'Options'.

Fig. Base de datos

### 6.2.9 Sprint retrospective

#### ¿Qué aprendimos durante el Sprint 2?

- La integración de MongoDB con el sistema de roles fue más sencilla de lo esperado gracias a una planificación anticipada.
- La correcta configuración de la base de datos es crucial para asegurar que todos los usuarios y administradores puedan acceder a sus respectivos permisos sin errores.
- La validación de roles a nivel de base de datos asegura la seguridad y la correcta asignación de privilegios a cada usuario, lo que reduce los riesgos de errores o accesos no autorizados.

#### ¿Qué hicimos bien?

- La base de datos MongoDB fue configurada sin problemas, lo que permitió un almacenamiento eficiente de los datos de los usuarios y sus roles.

- La integración de los roles con la base de datos se realizó de manera exitosa y sin conflictos importantes.
- El equipo logró coordinarse eficazmente en la configuración y pruebas de la base de datos, lo que facilitó la integración sin retrasos.

### **¿Qué podemos mejorar?**

- Asegurarnos de realizar más pruebas de validación de roles en diferentes escenarios para garantizar que no haya conflictos con los permisos en entornos reales.
- Mejorar la documentación técnica de la base de datos, especialmente sobre cómo se gestionan los roles y la interacción con la aplicación web.
- Realizar pruebas más exhaustivas con usuarios de diferentes roles para identificar posibles problemas en la gestión de permisos.

## **6.3 Desarrollo del Sprint 3**

### **6.3.1 Sprint planning**

#### **Objetivo del Sprint 3:**

- Desarrollar el módulo de creación de rutinas personalizadas para el Admin.
- Crear la funcionalidad para visualizar las rutinas creadas.
- Asegurar la correcta integración entre el sistema de usuarios, roles y las rutinas.

#### **Actividades planificadas:**

- Desarrollar interfaz para la creación de rutinas.

- Crear lógica para la asignación de rutinas a usuarios.
- Desarrollar backend para gestionar rutinas personalizadas.
- Validar que las rutinas se visualicen correctamente en el dashboard de los usuarios.

### 6.3.2 Sprint backlog

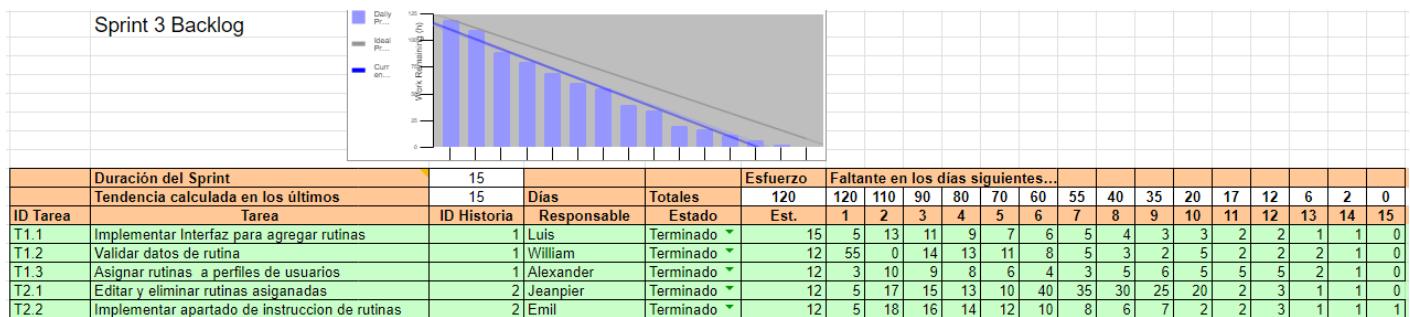


Fig. Sprint Backlog - Sprint 3

### 6.3.3 Historias de usuarios

ID Historia de Usuario	Como	Deseo	Para
HU007	Admin	Crear rutinas personalizadas	Ofrecer planes de ejercicios a los usuarios
HU008	Admin	Asignar rutinas a usuarios	Que los usuarios reciban rutinas personalizadas
HU009	Usuario	Ver el dashboard con información actualizada	Visualizar la cantidad de registros

Tabla. Historia de Usuarios Sprint 2

### 6.3.4 Taskboard

ID Tarea	Tarea	ID Historia
T1.1	Implementar Interfaz para agregar rutinas	7
T1.2	Validar datos de rutina	7
T1.3	Asignar rutinas a perfiles de usuarios	8
T2.1	Editar y eliminar rutinas asignadas	8
T2.2	Implementar apartado de instrucción de rutinas	9

## Fig. Taskboard- Sprint 3

### 6.3.5 Daily scrum

#### MINUTA DE REUNIÓN #3

**LUGAR:** Google Meet

**FECHA:** 14 de septiembre de 2024

**HORA:** 8:00 pm

**OBJETIVO:** Revisión del progreso de la creación y asignación de rutinas.

#### Participantes:

- Alcedo Muñoz Luis Fernando
- Navarro Arias Michael Brandon
- Nolasco Meza Nilton Anthony
- Paucarchuco Ramos Alexander David
- Ramirez Crispin William Alexis

#### Resumen de la reunión:

- El equipo ha avanzado en la creación de la interfaz para las rutinas.
- Se ha comenzado con la implementación del backend para la asignación de rutinas.
- Se ha discutido cómo mejorar la visualización de las rutinas en el dashboard del usuario.

#### Acuerdos:

- Finalizar la integración del backend de rutinas.
- Asegurar que las rutinas se muestren correctamente en el dashboard.

### 6.3.6 Sprint review

#### Historias de Usuario completadas en Sprint 3:

- **HU007:** Crear módulo de rutinas personalizadas.
- **HU008:** Asignar rutinas a usuarios.
- **HU009:** Visualizar rutinas en el dashboard.

#### Métricas clave:

- Historias de usuario completadas: 3/3 (100%)
- Retraso acumulado: 0 días

#### Próximos avances:

- Mejorar el módulo de rutinas personalizadas.
- Desarrollar las funcionalidades para dietas personalizadas para el Admin en el siguiente sprint.

### 6.3.7 Criterios de aceptación

ID HU	Descripción de la Tarea	Criterio de aceptacion
HU007	Crear rutinas personalizadas	El Admin puede crear, guardar y editar rutinas con al menos 3 ejercicios y una categoría asignada.
HU008	Asignar rutinas a usuarios	El Admin puede asignar, modificar o eliminar rutinas de usuarios, y el usuario recibe una notificación.
HU0069	Ver el dashboard con información actualizada	Visualizar la cantidad de registros

Tabla. Tabla de criterios de aceptación - Sprint 3

### 6.3.8 Resultados del sprint

#### 6.3.8.1 Evidencias.

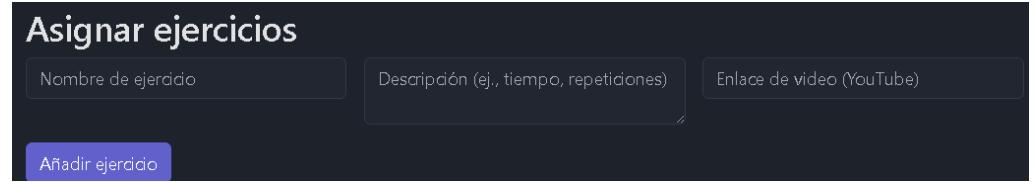


Fig. Asignación de ejercicios implementado

#### 6.3.8.2 Prueba de desarrollo.

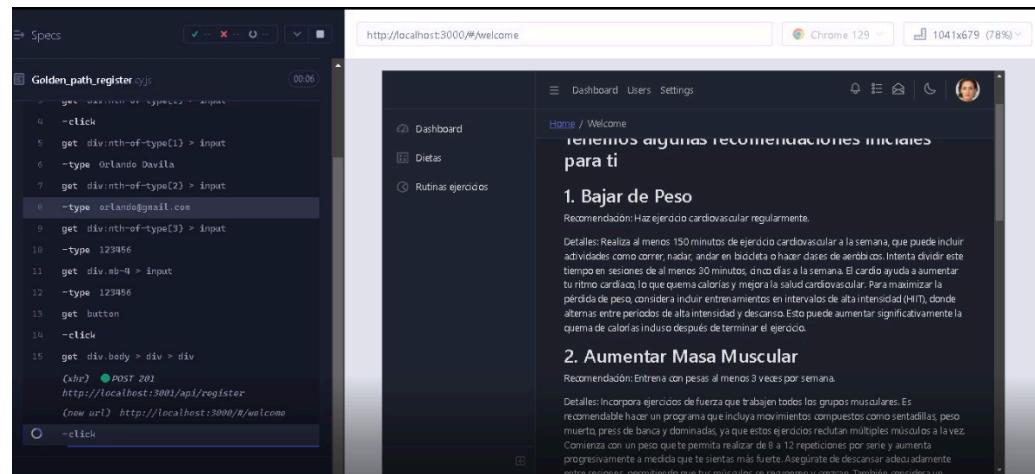


Fig. Prueba realizada en cypress

### 6.3.9 Sprint retrospective

#### ¿Qué aprendimos durante el Sprint 3?

- La importancia de tener una integración fluida entre MongoDB y la aplicación web para la gestión de usuarios y roles.
- El uso de un sistema de roles basado en MongoDB es esencial para la escalabilidad y flexibilidad del proyecto.

- Las pruebas de integración tempranas son fundamentales para detectar posibles inconsistencias.

### **¿Qué hicimos bien?**

- La base de datos fue configurada correctamente, y los roles fueron integrados de manera exitosa.
- La comunicación y la colaboración entre los miembros del equipo fueron efectivas durante la resolución de problemas técnicos relacionados con la base de datos.

### **¿Qué podemos mejorar?**

- Mejorar la documentación técnica sobre la integración de MongoDB para facilitar el trabajo de futuros desarrollos y la integración de otros módulos.
- Realizar pruebas de estrés para asegurar que el sistema pueda manejar un número elevado de usuarios sin errores.

## **6.4 Desarrollo del Sprint 4**

### **6.4.1 Sprint planning**

#### **Objetivo del Sprint 4:**

- Desarrollar y mejorar el módulo de creación de dietas personalizadas para el Admin.
- Permitir la asignación de dietas a usuarios y visualizar su progreso.

#### **Actividades planificadas:**

- Desarrollar la interfaz de dietas personalizadas.

- Crear backend para gestionar las dietas.
- Asignar dietas a usuarios y validación en el sistema.
- Mejorar la visualización de rutinas y dietas en el dashboard de los usuarios.

#### 6.4.2 Sprint backlog

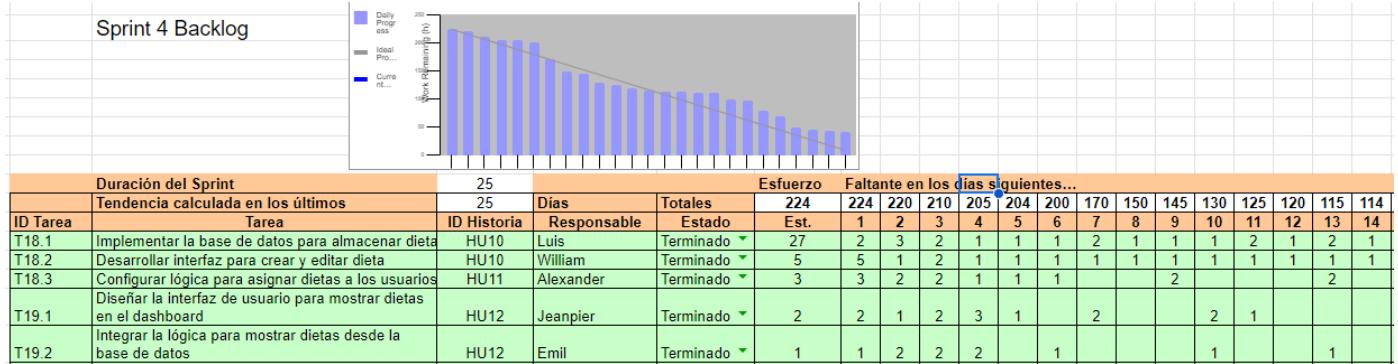


Fig. Sprint Backlog - Sprint 4

#### 6.4.3 Historias de usuarios

ID Historia de Usuario	Como	Deseo	Para
HU010	Admin	Crear dietas personalizadas	Ofrecer planes de alimentación a los usuarios
HU011	Admin	Asignar dietas a usuarios	Que los usuarios reciban dietas personalizadas
HU012	Usuario	Ver dietas en el dashboard	Visualizar la cantidad de registros

Tabla. Historia de Usuarios Sprint 2

#### 6.4.4 Taskboard

ID Tarea	Tarea	ID Historia
T18.1	Implementar la base de datos para almacenar dieta	HU10
T18.2	Desarrollar interfaz para crear y editar dieta	HU10
T18.3	Configurar lógica para asignar dietas a los usuarios	HU11
T19.1	Diseñar la interfaz de usuario para mostrar dietas en el dashboard	HU12
T19.2	Integrar la lógica para mostrar dietas desde la base de datos	HU12

Fig. Taskboard- Sprint 4

#### 6.4.5 Daily scrum

##### MINUTA DE REUNIÓN #4

**LUGAR:** Google Meet

**FECHA:** 30 de septiembre de 2024

**HORA:** 8:00 pm

**OBJETIVO:** Revisión de la creación y asignación de dietas personalizadas.

##### Participantes:

- Alcedo Muñoz Luis Fernando
- Navarro Arias Michael Brandon
- Nolasco Meza Nilton Anthony
- Paucarchuco Ramos Alexander David
- Ramirez Crispin William Alexis

##### Resumen de la reunión:

- El equipo ha avanzado en la interfaz de creación de dietas.
- El backend para la asignación de dietas ya está operativo.
- Se está trabajando en la visualización de dietas en el dashboard del usuario.

**Acuerdos:**

- Finalizar la integración de dietas personalizadas.
- Realizar pruebas de asignación de dietas a usuarios.

**6.4.6 Sprint review**

**Historias de Usuario completadas en Sprint 4:**

- **HU010:** Crear módulo de dietas personalizadas.
- **HU011:** Asignar dietas a usuarios.
- **HU012:** Visualizar dietas en el dashboard.

**Métricas clave:**

- Historias de usuario completadas: 3/3 (100%)
- Retraso acumulado: 0 días

**Próximos avances:**

- Realizar el lanzamiento de la web.

**6.4.7 Criterios de aceptación**

ID HU	Descripción de la Tarea	Descripción de la Tarea
HU010	Desarrollar interfaz para la creación de dietas	La interfaz debe permitir al admin crear, editar y eliminar dietas personalizadas.
HU011	Crear backend para asignación de dietas	El backend debe asignar dietas correctamente a los usuarios.
HU012	Implementar visualización de dietas en el dashboard	Visualizar la cantidad de registros

Tabla. Tabla de criterios de aceptación - Sprint 4

#### 6.4.8 Resultados del sprint

##### 6.4.8.1 Evidencias.



#	Nombre	Email	Acciones
1	Entrenador	entrenador@gmail.com	<button>Editar</button> <button>Eliminar</button> <button>Asignar Rutina</button> <button>Asignar Dieta</button>
2	marco	marco@gmail.com	<button>Editar</button> <button>Eliminar</button> <button>Asignar Rutina</button> <button>Asignar Dieta</button>
3	William	william@gmail.com	<button>Editar</button> <button>Eliminar</button> <button>Asignar Rutina</button> <button>Asignar Dieta</button>

Fig. Edición de usuarios Implementado

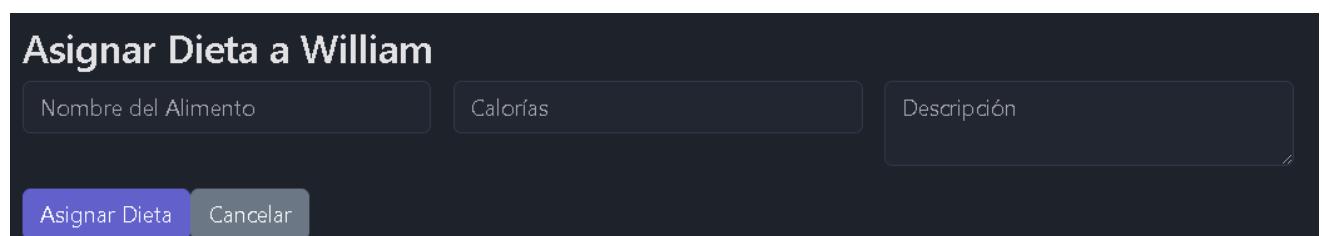


Fig. Asignación de Dieta Implementado

#### 6.4.8.1 Prueba de desarrollo.

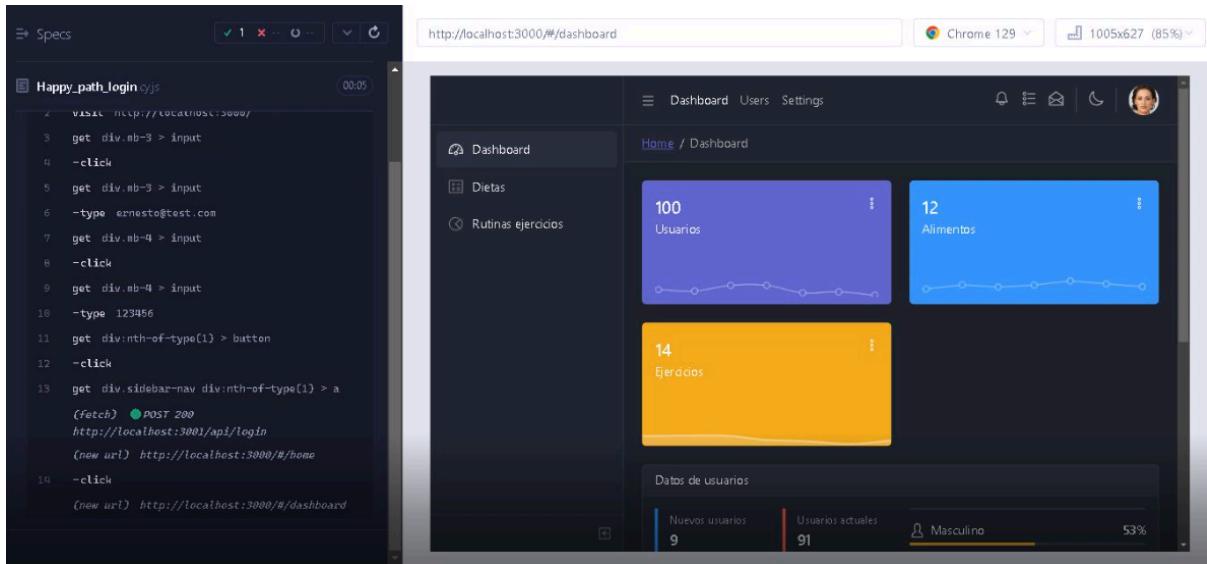


Fig. Prueba realizada en cypress

#### 6.4.9 Sprint retrospective

##### ¿Qué aprendimos durante el Sprint 4?

- La importancia de la integración entre la interfaz de usuario (frontend) y el backend para la asignación de dietas. Fue crucial que ambos sistemas se comuniquen correctamente para que el proceso fuera fluido.
- El valor de las pruebas unitarias y de integración al trabajar con bases de datos, ya que nos permitió asegurarnos de que las dietas asignadas se guardaban correctamente y eran accesibles por los usuarios.

##### ¿Qué hicimos bien?

- La implementación de la interfaz para la creación de dietas fue bien recibida y se completó sin mayores complicaciones.

- La asignación de dietas a los usuarios se realizó sin errores, y la validación en la base de datos se completó correctamente.

### ¿Qué podemos mejorar?

- Mejorar la visualización en el dashboard, asegurándonos de que las dietas sean presentadas de manera clara y que se pueda acceder fácilmente a los detalles de cada dieta.
- Agregar más validaciones en el backend para evitar posibles errores de asignación de dietas a usuarios.

## CAPÍTULO 7

### PRUEBAS DE SOFTWARE

#### 7.1. Plan de Pruebas

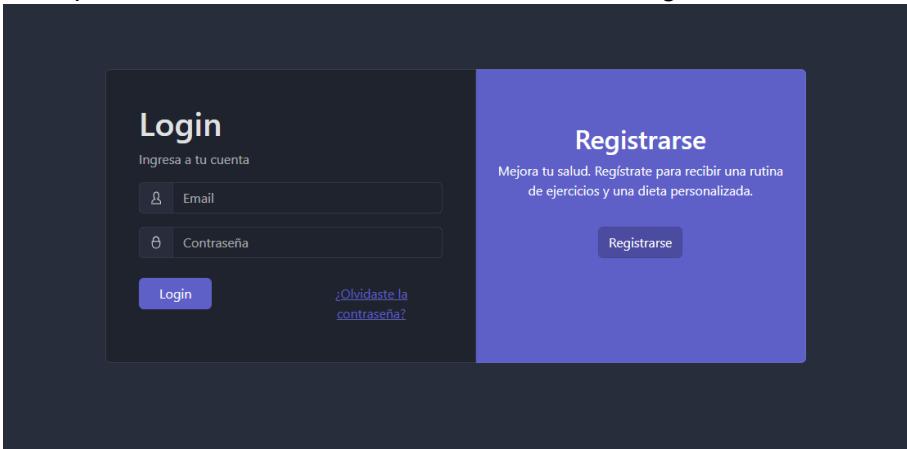
El propósito de utilizar Selenium para pruebas de software es asegurar la calidad y el rendimiento de las aplicaciones web mediante la automatización. Estas pruebas permiten verificar la funcionalidad, garantizar la compatibilidad con diferentes navegadores y detectar posibles regresiones. Este enfoque incrementa la eficiencia, reduce el tiempo necesario para las pruebas y contribuye a un software de mayor calidad al identificar y solucionar problemas tempranamente en el proceso de desarrollo.

Configuración del entorno:

- **Instalar Selenium IDE:** Añade la extensión de Selenium IDE al navegador Chrome.
- **Abrir la herramienta:** Una vez instalada, accede a Selenium IDE desde su ícono ubicado en la barra de herramientas de Chrome.
- **Configurar el entorno:** Una vez abierto Selenium IDE, puedes comenzar a crear pruebas automatizadas grabando las acciones realizadas durante una sesión de navegación web y reproduciéndolas posteriormente como pruebas.

- **Escribir y modificar pruebas:** Además de grabar acciones, tienes la opción de redactar pruebas manualmente utilizando el lenguaje de scripting propio de Selenium IDE. Estas pruebas se pueden ajustar y personalizar según las necesidades específicas.
- **Ejecutar pruebas:** Una vez creadas, puedes ejecutar las pruebas directamente en Selenium IDE para validar su funcionamiento.
- **Guardar y organizar pruebas:** Las pruebas creadas se pueden almacenar dentro de Selenium IDE para su reutilización futura. Además, es posible organizarlas en conjuntos o suites para facilitar su administración.

### **SPRINT 1: Establecer la funcionalidad básica del sistema enfocándose en el registro de usuarios, inicio de sesión y asignación de roles.**

<b>HU001: PRUEBA DE SISTEMA - Registrarme en la plataforma. (Rol: Usuario)</b>	
<b>Descripción</b>	La prueba de funcionamiento consiste en que se cumpla el registro de datos solicitados con éxito al momento de registrarse.
<b>Precondiciones</b>	- Datos del usuario.
<b>Pasos</b>	<p>En la pantalla de inicio, seleccionamos el botón Registrarse:</p>  <p>Ya en la ventana de Registro, rellenamos los datos solicitado y luego hacemos click en “Crear cuenta”:</p>

# Registro

Crea tu cuenta

PRUEBA

prueba1@gamil.com

...

...

**Crear cuenta**

De esta manera podemos, ingresar al menú del sistema:

¡Registro exitoso! Bienvenido.

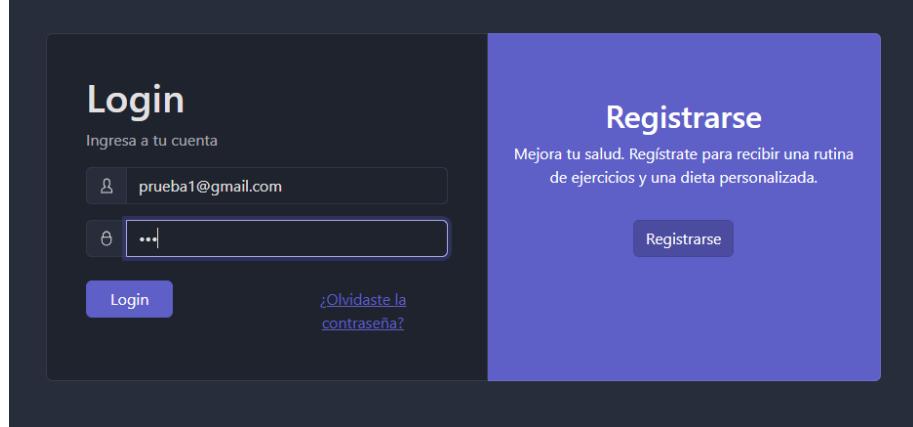
Tenemos algunas recomendaciones iniciales para ti

- 1. Bajar de Peso**  
Recomendación: Haz ejercicio cardiovascular regularmente.  
Detalles: Realiza al menos 150 minutos de ejercicio cardiovascular a la semana, que puede incluir actividades como correr, nadar, andar en bicicleta o hacer clases de aeróbicos. Intenta dividir este tiempo en sesiones de al menos 30 minutos, cinco días a la semana. El cardio ayuda a aumentar tu ritmo cardíaco, lo que quema calorías y mejora la salud cardiovascular. Para maximizar la pérdida de peso, considera incluir entrenamientos en intervalos de alta intensidad (HIIT), donde alternas entre períodos de alta intensidad y descanso. Esto puede aumentar significativamente la quema de calorías incluso después de terminar el ejercicio.
- 2. Aumentar Masa Muscular**  
Recomendación: Entrena con pesas al menos 3 veces por semana.

<b>Resultados Esperados</b>	El registro del usuario y creación de la cuenta sean exitosos.
<b>Criterios de Validación</b>	<ul style="list-style-type: none"> <li>- Confirmación de la creación de la cuenta.</li> <li>- Los datos ingresados coinciden con los datos guardados.</li> </ul>
<b>Resultado</b>	Usuario registrado exitosamente.

#### HU002: PRUEBA DE SISTEMA - Iniciar sesión en mi cuenta. (Rol: Usuario)

<b>Descripción</b>	La prueba funcional consiste en poder iniciar sesión con alguna cuenta recién registrada y/o ya registrada.
<b>Precondiciones</b>	<ul style="list-style-type: none"> <li>- El usuario debe registrarse y crear su cuenta.</li> <li>- Los datos registrados coinciden con los datos a ingresar.</li> </ul>

<b>Pasos</b>	<p>Otra vez en la ventana de Inicio, debemos llenar las casillas Email y Contraseña, luego hacer click en el botón “Login”:</p>  <p>De esta manera, podemos ingresar a la venta principal:</p> 
<b>Resultados Esperados</b>	El inicio de sesión sea exitoso.
<b>Criterios de Validación</b>	Los datos de usuario y contraseña tengan la verificación de coincidir con los datos guardados.
<b>Resultado</b>	Se inició sesión correctamente.

HU003: PRUEBA DE SISTEMA - Otorgar roles y permisos a los usuarios. (Rol: Administrador)	
<b>Descripción</b>	La prueba funcional de este caso consiste en poder dar roles y permisos a los usuarios, ya sean clientes o administradores (personal del gimnasio).
<b>Precondiciones</b>	<ul style="list-style-type: none"> <li>- Ser un usuario registrado.</li> <li>- Iniciar sesión satisfactoriamente.</li> </ul>
<b>Pasos</b>	De la ventana principal, ingresamos a la ventana Dashboard Admin:



Rellenamos los datos, y en la parte de Rol, seleccionamos Administrador, luego click en “Añadir”:

#	Nombre	Email	Acciones
1	prueba	prueba@gmail.com	<a href="#">Editar</a> <a href="#">Eliminar</a> <a href="#">Asignar Rutina</a> <a href="#">Asignar Dieta</a>
2	PRUEBA	prueba1@gmail.com	<a href="#">Editar</a> <a href="#">Eliminar</a> <a href="#">Asignar Rutina</a> <a href="#">Asignar Dieta</a>
3	prueba	administrador1@gmail.com	<a href="#">Editar</a> <a href="#">Eliminar</a> <a href="#">Asignar Rutina</a> <a href="#">Asignar Dieta</a>

Y en la lista de Usuarios podemos ver el nuevo usuario con permisos de Administrador:

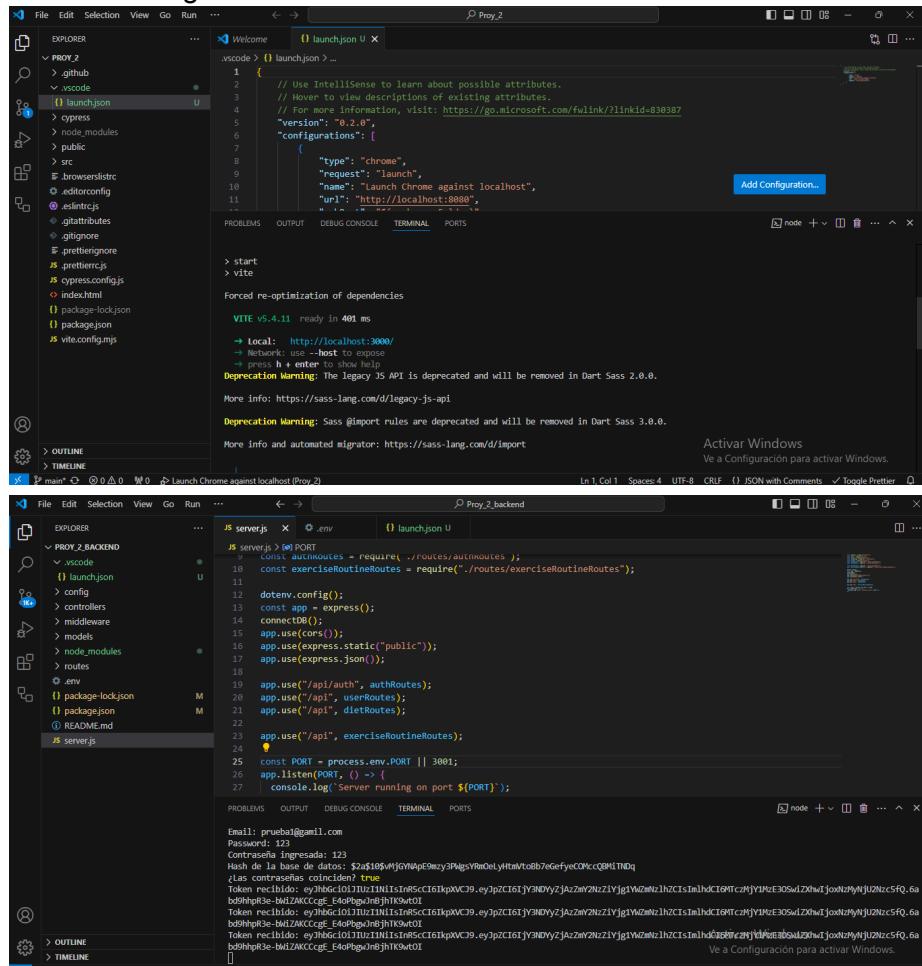
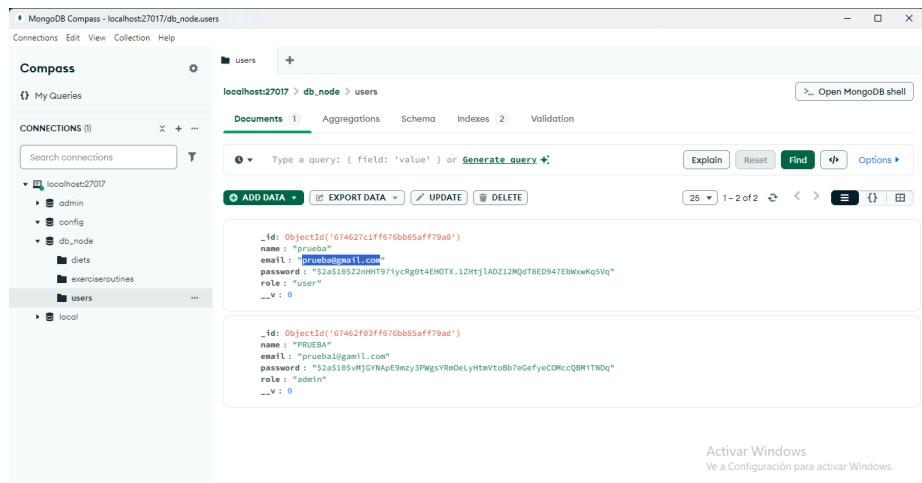
#	Nombre	Email	Acciones
1	prueba	prueba@gmail.com	<a href="#">Editar</a> <a href="#">Eliminar</a> <a href="#">Asignar Rutina</a> <a href="#">Asignar Dieta</a>
2	PRUEBA	prueba1@gmail.com	<a href="#">Editar</a> <a href="#">Eliminar</a> <a href="#">Asignar Rutina</a> <a href="#">Asignar Dieta</a>
3	prueba	administrador1@gmail.com	<a href="#">Editar</a> <a href="#">Eliminar</a> <a href="#">Asignar Rutina</a> <a href="#">Asignar Dieta</a>

<b>Resultados Esperados</b>	Cada usuario cuenta con permisos definitivos en el sistema y un rol.
<b>Criterios de Validación</b>	<ul style="list-style-type: none"> <li>- Un usuario con cuenta de administrador pueda otorgar los permisos y roles.</li> <li>- El usuario debe ser alguien registrado y con una cuenta activa.</li> </ul>
<b>Resultado</b>	Nuevo usuario con el rol de Administrador.

## SPRINT 2: Integrar la base de datos MongoDB con el sistema para gestionar usuarios y roles.

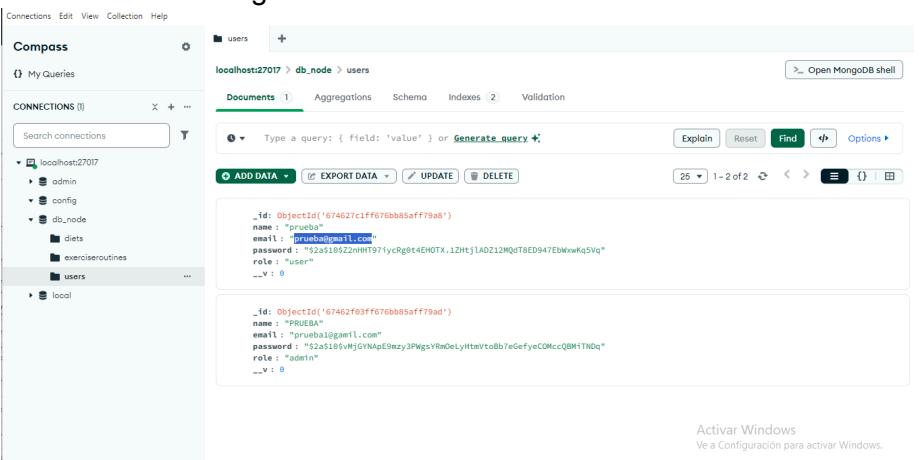
**HU004: PRUEBA DE SISTEMA - Gestionar base de datos de usuarios. (Rol: Administrador)**

<b>Descripción</b>	La prueba funcional en este caso verificará que el sistema cuenta con una conexión estable con la base de datos MongoDB.
--------------------	--

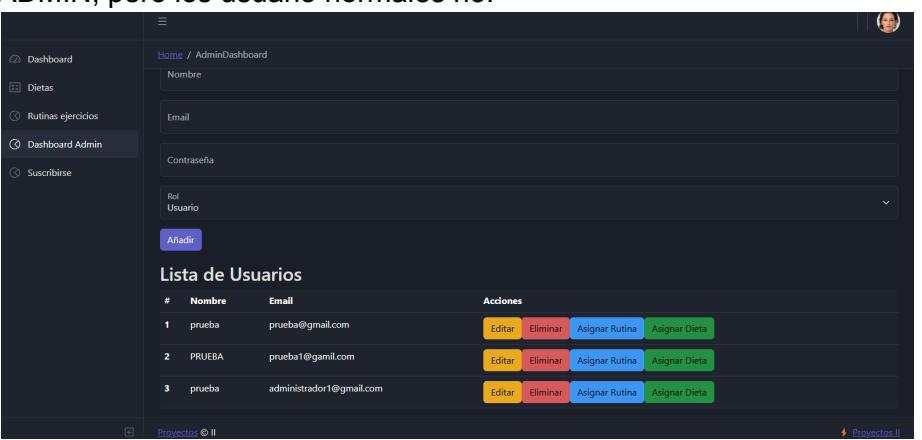
<b>Precondiciones</b>	<ul style="list-style-type: none"> <li>- Configuración correcta de la base de datos MongoDB.</li> <li>- Conexión predeterminada entre la aplicación y la base de datos.</li> </ul>
<b>Pasos</b>	<p>Conexión correcta por parte del código con npm y node con la base de datos MongoDB:</p>  <pre>launch.json {     "version": "0.2.0",     "configurations": [         {             "type": "chrome",             "request": "launch",             "name": "Launch Chrome against localhost",             "url": "http://localhost:8889",             "startFile": "\${workspaceRoot}/src/index.html"         }     ] }  server.js const PORT = process.env.PORT    3001; app.listen(PORT, () =&gt; {     console.log(`Server running on port \${PORT}`); })</pre>
<b>Resultados Esperados</b>	<p>La conexión entre la base de datos y la aplicación sea estable y rápida.</p>  <pre>users {     "_id": ObjectID("67462f03ff670bb85aff79ad"),     "name": "PRUEBA",     "email": "prueba@gmail.com",     "password": "52a51e03220HHT97ycRg0t4EHOTX.1Zhtj1ADz12MqdTBED947EbKwKqB1TNDq",     "role": "user",     "__v": 0 }</pre>

<b>Criterios de Validación</b>	Los datos registrados se puedan observar en el MongoDB.
<b>Resultado</b>	La conexión entre el sistema y la base de datos es estable.

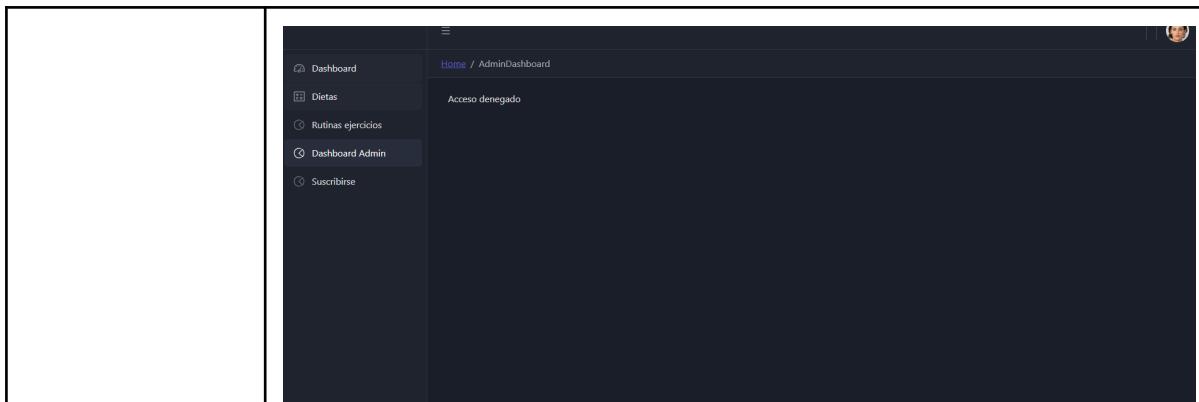
#### HU005: PRUEBA DE SISTEMA - Vincular roles con la base de datos. (Rol: Usuario)

<b>Descripción</b>	La prueba funcional para este caso los usuarios deben observar los permisos otorgados ya sean administrador o cliente.
<b>Precondiciones</b>	<ul style="list-style-type: none"> <li>- El usuario debe contar con una cuenta registrada.</li> <li>- Debe poder iniciar sesión.</li> </ul>
<b>Pasos</b>	<p>Nos dirigimos al software MongoDB, y nos vamos a la subcarpeta db_node, luego en users, de esta manera podemos verificar que los datos están siendo guardados satisfactoriamente:</p> 

Por parte del sistema respecto a los roles los usuarios administradores deben poder ingresar a la interfaz DASHBOARD ADMIN, pero los usuarios normales no:



(Interfaz Cuenta Administrativa)



(Interfaz de una Cuenta de usuario normal).

<b>Resultados Esperados</b>	El usuario debe poder observar ciertos datos como también debe tener permisos para ciertas acciones en el sistema.
<b>Criterios de Validación</b>	Los usuarios deben realizar ciertas acciones y poder solo ver ciertos datos.
<b>Resultado</b>	Se muestra correctamente el impedimento en el ingreso a usuarios administrativos a diferencia de usuarios normales.

#### HU006: PRUEBA DE SISTEMA - Validar roles con base de datos. (Rol: Administrador)

<b>Descripción</b>	En esta prueba funcional se podrá observar y verificar la asignación de roles y permisos correctos a cada usuario.
<b>Precondiciones</b>	- Usuario registrado.
<b>Pasos</b>	Para poder verificar los roles de los usuarios en la base de datos, vamos a db_node, luego a users y podemos ver los datos de los usuarios y sus roles como user o admin: 
<b>Resultados Esperados</b>	El sistema debe mostrar los roles y permisos otorgados a cada usuario.
<b>Criterios de Validación</b>	El sistema debe mostrar los roles y permisos de cada usuario, que coincidan con la base de datos.

<b>Resultado</b>	La base de datos coincide con los registros o cambios realizados a los roles de los usuarios.
------------------	---

### SPRINT 3: Implementar la funcionalidad para crear y asignar rutinas personalizadas a los usuarios.

<b>HU007: PRUEBA DE SISTEMA - Crear rutinas personalizadas. (Rol: Administrador)</b>	
<b>Descripción</b>	La prueba funcional aquí permitirá que el administrador pueda crear rutinas personalizadas.
<b>Precondiciones</b>	- El administrador debe tener cuenta propia con los permisos necesarios.
<b>Pasos</b>	<p>De la ventana principal, ingresamos al dashboard admin, luego en cualquiera de los usuarios daremos click en el botón “Asignar rutina”:</p> <p>Luego daremos click en el botón “Abrir perfil”:</p> <p>Ya abierto el perfil, rellenamos las casillas solicitadas y daremos click en “Añadir ejercicio”:</p>

The screenshot shows the Admin Dashboard interface. On the left sidebar, there are navigation links: Dashboard, Dietas, Rutinas ejercicios, Dashboard Admin, and Suscribirse. The main content area has a header "Home / AdminDashboard". Below it is a section titled "Lista de Usuarios" (User List) with a table showing three users:

#	Nombre	Email	Acciones			
1	prueba	prueba@gmail.com	<a href="#">Editar</a>	<a href="#">Eliminar</a>	<a href="#">Asignar Rutina</a>	<a href="#">Asignar Dieta</a>
2	PRUEBA	prueba1@gmail.com	<a href="#">Editar</a>	<a href="#">Eliminar</a>	<a href="#">Asignar Rutina</a>	<a href="#">Asignar Dieta</a>
3	prueba	administrador1@gmail.com	<a href="#">Editar</a>	<a href="#">Eliminar</a>	<a href="#">Asignar Rutina</a>	<a href="#">Asignar Dieta</a>

Below this is a "Perfil de prueba" (Profile of prueba) section with a "Cerrar perfil" (Close profile) button. The "Asignar ejercicios" (Assign Exercises) section contains a form for adding exercises. It has fields for "Nombre de ejercicio" (Exercise name), "Descripción (ej., tiempo, repeticiones)" (Description (e.g., time, repetitions)), and "Enlace de video (YouTube)" (YouTube video link). A preview of the exercise "SENTADILLAS" is shown with its description: "REALIZAR SENTADILLA SIN PESO EXTRA. 4 SERIES DE 12 REPETICIONES." and a YouTube link: "https://www.youtube.com/watch?v=BjixzWEw4EY". There is also a "Añadir ejercicio" (Add exercise) button.

Ahora podemos ver que el ejercicio nuevo fue agregado:

This screenshot shows the same Admin Dashboard interface as the previous one, but with a new entry in the exercise assignment section. The "Asignar ejercicios" section now includes a list of exercises with their descriptions and YouTube links. The new entry is ". SENTADILLAS" with the description "REALIZAR SENTADILLA SIN PESO EXTRA. 4 SERIES DE 12 REPETICIONES." and the same YouTube link. It also has its own "Editar" and "Eliminar" buttons.

<b>Resultados Esperados</b>	El administrador debe poder generar rutinas satisfactoriamente en el sistema.
<b>Criterios de Validación</b>	<ul style="list-style-type: none"> <li>- El sistema debe poder mostrar las rutinas creadas y verificar que se le fue asignado al usuario.</li> </ul>
<b>Resultado</b>	La rutina fue creada con éxito.

#### HU008: PRUEBA DE SISTEMA - Asignar rutinas a usuarios. (Rol: Administrador)

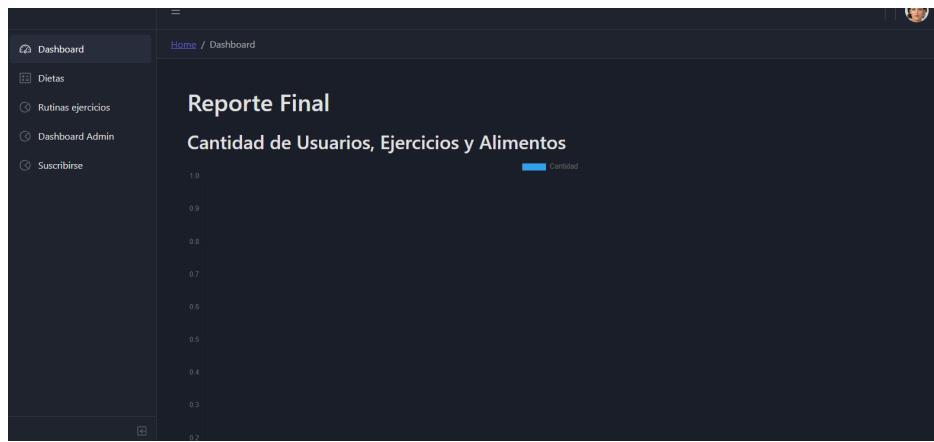
<b>Descripción</b>	El administrador debe poder asignar las rutinas creadas a los usuarios.
<b>Precondiciones</b>	<ul style="list-style-type: none"> <li>- Rutina creada.</li> </ul>
<b>Pasos</b>	Del punto anterior donde se creó el ejercicio, también ya se fue asignado el ejercicio al usuario seleccionado:

Ahora por la ventana del usuario prueba, podemos ver en su ventana “Rutinas ejercicios”, el ejercicio recién creado y asignado:

<b>Resultados Esperados</b>	El administrador asigna las rutinas a los usuarios y estas deben aparecer en la ventana de ejercicios de los usuarios.
<b>Criterios de Validación</b>	La serie de ejercicios deben mostrarse tal como se creó en el sistema de los usuarios.
<b>Resultado</b>	El ejercicio fue asignado correctamente al usuario.

#### **HU009: PRUEBA DE SISTEMA - Ver el dashboard con información actualizada. (Rol: Usuario)**

<b>Descripción</b>	El usuario y/o el administrador debe poder ver los avances de los usuarios respecto a la finalización de sus ejercicios por día.
<b>Precondiciones</b>	<ul style="list-style-type: none"> <li>- Rutina creada.</li> <li>- Rutina asignada.</li> <li>- Usuario debe cumplir con sus rutinas.</li> </ul>
<b>Pasos</b>	Para poder ver nuestro avance o desarrollo en del Dashboard debemos ingresar a la ventana Dashboard:

	 
Resultados Esperados	El usuario debe poder ver sus avances actualizados en el Dashboard, según su desarrollo propio.
Criterios de Validación	<ul style="list-style-type: none"> <li>- El Dashboard debe presentar datos actualizados.</li> <li>- Coincidir con los avances de los usuarios en sus ejercicios.</li> </ul>
Resultado	Se puede observar el dashboard personal de cada usuario.

#### **SPRINT 4: Desarrollar el módulo de dietas personalizadas y mejorar la interacción en el Dashboard.**

<b>HU010: PRUEBA DE SISTEMA - Crear dietas personalizadas. (Rol: Administrador)</b>	
<b>Descripción</b>	El administrador o el personal del gimnasio debe crear dietas satisfactoriamente.
<b>Precondiciones</b>	<ul style="list-style-type: none"> <li>- El usuario debe tener los permisos necesarios.</li> <li>- El usuario debe estar registrado.</li> </ul>
<b>Pasos</b>	Vamos a la ventana de Dashboard admin, y hacemos click en Asignar Dieta en uno de los usuarios:

--	--

Luego rellenamos las casillas y daremos click en Asignar Dieta:

--	--

<b>Resultados Esperados</b>	El entrenador o nutricionista debe poder crear dietas personalizadas para cada usuario.
<b>Criterios de Validación</b>	Creación y registro de las dietas personalizadas.
<b>Resultado</b>	De esta manera se pudo crear la dieta para el usuario.

<b>HU011: PRUEBA DE SISTEMA - Asignar dietas a usuarios. (Rol: Administrador)</b>	
<b>Descripción</b>	El nutricionista o administrador debe poder asignar las dietas creadas a los usuarios respectivamente.
<b>Precondiciones</b>	<ul style="list-style-type: none"> <li>- Usuario registrado.</li> <li>- Dieta creada.</li> <li>- Debe contar con permisos de administrador.</li> </ul>
<b>Pasos</b>	Con los pasos anteriores ya se pudo asignar la dieta al usuario.

<b>Resultados Esperados</b>	El administrador debe poder asignar distintas dietas a los usuarios.
<b>Criterios de Validación</b>	<ul style="list-style-type: none"> <li>- El sistema debe poder verificar que la dieta fue asignada al usuario.</li> <li>- El usuario debe poder ver la dieta asignada.</li> </ul>
<b>Resultado</b>	La dieta fue asignada al usuario.

<b>HU012: PRUEBA DE SISTEMA - Ver dietas en el dashboard. (Rol: Usuario).</b>	
<b>Descripción</b>	La dieta asignada al usuario debe poder verse en el Dashboard personal.
<b>Precondiciones</b>	<ul style="list-style-type: none"> <li>- Dieta asignada.</li> <li>- Dieta creada.</li> <li>- Usuario registrado.</li> </ul>
<b>Pasos</b>	<p>Para poder ver nuestro avance o desarrollo en del Dashboard debemos ingresar a la ventana Dashboard:</p>

	
<b>Resultados Esperados</b>	El usuario debe poder observar su dieta personalizada en el Dashboard y cualquier actualización o cambio en esta debe ser actualizada inmediatamente.
<b>Criterios de Validación</b>	<ul style="list-style-type: none"> <li>- La dieta debe coincidir con la asignada por el administrador.</li> <li>- Debe mostrarse de manera actualizada a tiempo real.</li> </ul>
<b>Resultado</b>	El dashboard actualizará cuando se cumpla con la dieta.

## 7.2 Plan de Pruebas de Extremo a Extremo

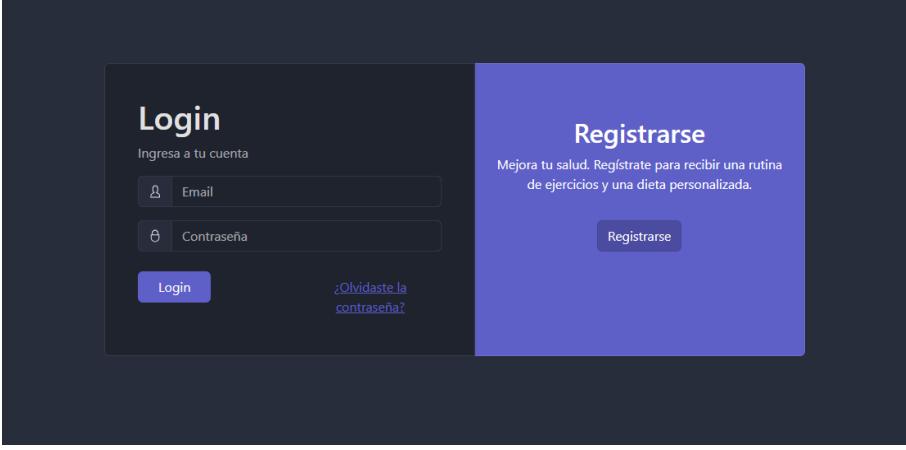
**Propósito:** El objetivo principal de este plan es comprobar que todas las funcionalidades del sistema de gestión de informes operen correctamente desde el inicio hasta el final de un flujo de usuario. Para lograrlo, se empleará Cypress, una herramienta diseñada para pruebas automatizadas en aplicaciones web, asegurando así la calidad del sistema.

### Preparación del Entorno:

- **Instalar Node.js:** Descarga e instala la versión más reciente desde la página oficial de Node.js ([nodejs.org](https://nodejs.org)).
- **Configurar el proyecto en VSCode:** Accede al directorio del proyecto desde Visual Studio Code (VSCode) y abre una terminal integrada.
- **Iniciar el proyecto con npm:** En la terminal, ejecuta el comando `npm init` para crear un nuevo proyecto de Node.js.
- **Instalar Cypress:** Ejecuta `npm install cypress --save-dev` en la terminal para añadir Cypress como una dependencia de desarrollo.
- **Configurar Cypress:** Una vez instalado, inicia Cypress por primera vez con `npx cypress open`. Esto generará automáticamente la estructura necesaria de directorios y subcarpetas, como fixtures, integration, plugins y support dentro de una carpeta llamada cypress.
- **Actualizar scripts en package.json:** Abre el archivo `package.json` y añade un script que permita ejecutar Cypress fácilmente.

- **Crear casos de prueba:** Dentro de la carpeta cypress/integration, crea archivos con extensión .spec.js. En estos archivos se desarrollarán los casos de prueba que validarán la funcionalidad de extremo a extremo.
- **Ejecutar las pruebas:** Usa la interfaz gráfica de Cypress para correr las pruebas de forma interactiva ejecutando npm run cypress:open. Alternativamente, para ejecutar las pruebas en modo headless (sin interfaz gráfica), utiliza el comando npm run cypress:run.

### **SPRINT 1:**

<b>HU001: Registrarme en la plataforma</b>	
<b>Descripción:</b> Verificar que los datos ingresados por el usuario se registren correctamente al crear una cuenta en la plataforma.	
<b>Pasos:</b>	<ul style="list-style-type: none"> <li>● Acceder a la plataforma de registro.</li> <li>● Completar el formulario con los datos solicitados.</li> <li>● Enviar la solicitud de registro.</li> <li>● Confirmar la creación de la cuenta.</li> </ul>
<b>Resultados Esperados:</b> El usuario debe poder registrarse satisfactoriamente, y la cuenta debe ser creada con los datos ingresados.	
	

# Registro

Crea tu cuenta

 PRUEBA

 prueba1@gamil.com

 ...

 ...

**Crear cuenta**

≡

Home / Welcome



 Dashboard

 Dietas

 Rutinas ejercicios

 Dashboard Admin

 Suscribirse

**¡Registro exitoso! Bienvenido.**

Tenemos algunas recomendaciones iniciales para ti

**1. Bajar de Peso**

Recomendación: Haz ejercicio cardiovascular regularmente.

Detalles: Realiza al menos 150 minutos de ejercicio cardiovascular a la semana, que puede incluir actividades como correr, nadar, andar en bicicleta o hacer clases de aeróbicos. Intenta dividir este tiempo en sesiones de al menos 30 minutos, cinco días a la semana. El cardio ayuda a aumentar tu ritmo cardíaco, lo que quema calorías y mejora la salud cardiovascular. Para maximizar la pérdida de peso, considera incluir entrenamientos en intervalos de alta intensidad (HIIT), donde alternas entre períodos de alta intensidad y descanso. Esto puede aumentar significativamente la quema de calorías incluso después de terminar el ejercicio.

**2. Aumentar Masa Muscular**

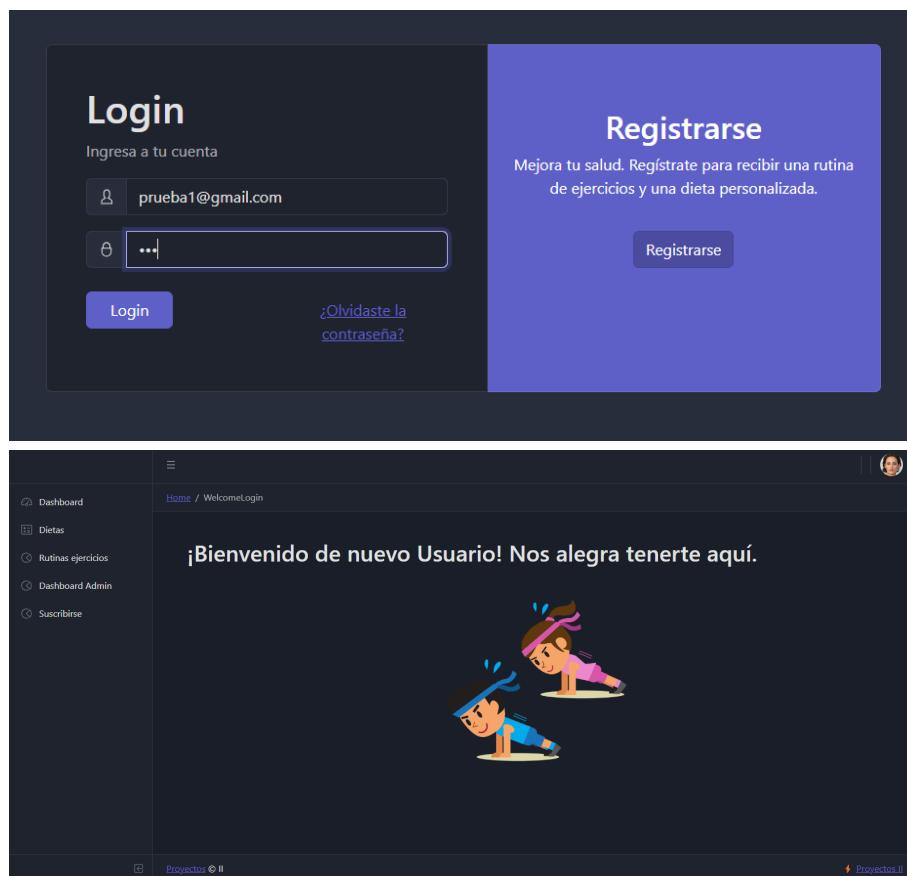
Recomendación: Entrena con pesas al menos 3 veces por semana.

## HU002: Iniciar sesión en mi cuenta

**Descripción:** Validar que un usuario pueda iniciar sesión con una cuenta previamente registrada.

- |               |   |
|---------------|---|
| <b>Pasos:</b> | <ul style="list-style-type: none"><li>• Acceder a la página de inicio de sesión.</li><li>• Ingresar las credenciales del usuario (correo y contraseña).</li><li>• Presionar el botón de inicio de sesión.</li></ul> |
|---------------|---|

**Resultados Esperados:** El usuario debe poder registrarse satisfactoriamente, y la cuenta debe ser creada con los datos ingresados.



## HU003: Definir los roles de usuario y admin

**Descripción:** Confirmar que un administrador pueda asignar roles y permisos a los usuarios registrados.

- |               |   |
|---------------|---|
| <b>Pasos:</b> | <ul style="list-style-type: none"><li>• Iniciar sesión como administrador.</li><li>• Navegar al módulo de gestión de roles y permisos.</li><li>• Seleccionar un usuario y asignarle un rol.</li><li>• Guardar los cambios realizados.</li></ul> |
|---------------|---|

**Resultados Esperados:** Cada usuario debe tener roles y permisos correctamente asignados según lo configurado por el administrador.

## SPRINT 2:

### HU004: Gestionar base de datos de usuarios

**Descripción:** Comprobar la conexión estable entre la aplicación y la base de datos MongoDB para gestionar usuarios.

<b>Pasos:</b>
---------------

- Configurar correctamente MongoDB.
- Establecer la conexión predeterminada entre la aplicación y la base de datos.
- Realizar una prueba de inserción de datos en MongoDB.

**Resultados Esperados:** La conexión debe ser estable, y los datos ingresados deben almacenarse en la base de datos correctamente.

The screenshot displays a developer's environment with three main components:

- VS Code (Top Window):** Shows the `launch.json` configuration file for a project named "Proy\_2". The configuration includes a single launch configuration for a Chrome browser against localhost.
- VS Code (Second Window):** Shows the `server.js` file for the "PROY\_2\_BACKEND" project. The file defines an Express app, sets up routes for auth, user, and diet, and starts the server on port 3001.
- MongoDB Compass (Bottom Window):** Connected to the "db\_node" database at "localhost:27017". The "users" collection is selected. A query is being run to find documents where the field "value" is present. Two documents are shown in the results table.

```

// Use IntelliSense to learn about possible attributes.
// Hover to view descriptions of existing attributes.
// For more information, visit: https://go.microsoft.com/fwlink/?LinkId=830387
{
  "version": "0.2.0",
  "configurations": [
    {
      "type": "chrome",
      "request": "launch",
      "name": "Launch Chrome against localhost",
      "url": "http://localhost:8080"
    }
  ]
}

// local: http://localhost:3000/
// Network: use --host to expose
// press h + enter to show help
Deprecation Warning: The Legacy JS API is deprecated and will be removed in Dart Sass 2.0.0.
More info: https://sass-lang.com/d/legacy-js-api
Deprecation Warning: Sass @import rules are deprecated and will be removed in Dart Sass 3.0.0.
More info and automated migrator: https://sass-lang.com/d/import

```

```

const autoroutes = require( './routes/autoroutes' );
const exerciseRoutineRoutes = require( './routes/exerciseRoutineRoutes' );
const dotenvConfig = require( './env' );
const app = express();
app.use(cors());
app.use(express.static("public"));
app.use(express.json());
app.use("/api/auth", authRoutes);
app.use("/api", userRoutes);
app.use("/api", dietRoutes);
app.use("/api", exerciseRoutineRoutes);
const PORT = process.env.PORT || 3001;
app.listen(PORT, () => {
  console.log(`Server running on port ${PORT}`);
}

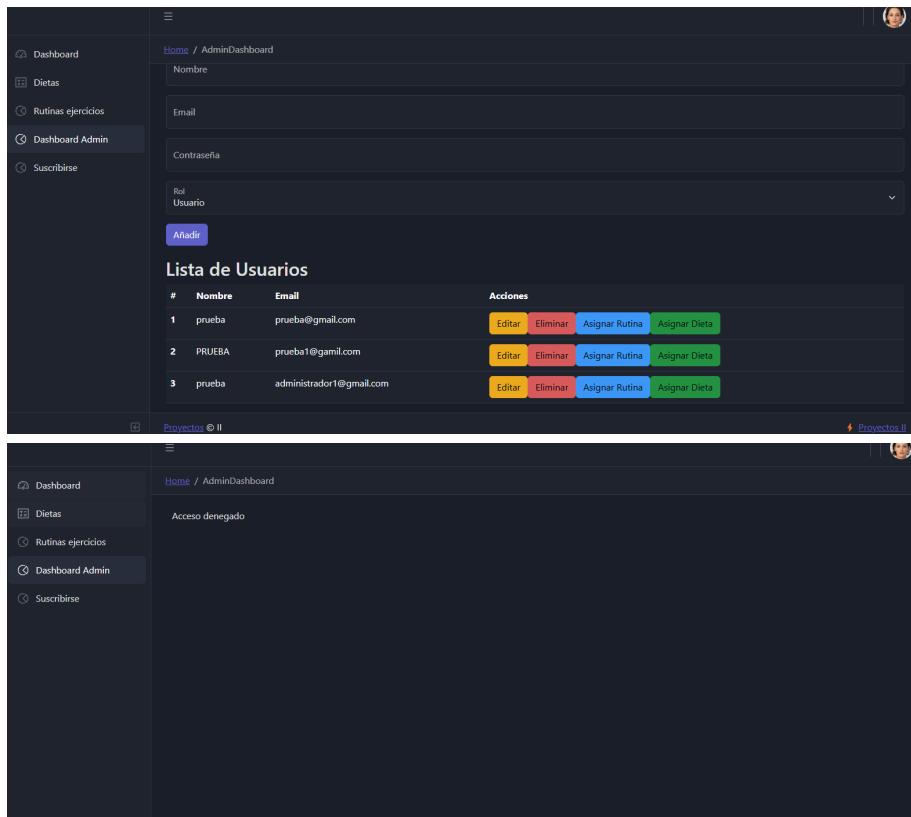
```

## HU005: Vincular roles con la base de datos

**Descripción:** Validar que los roles de los usuarios estén vinculados correctamente con la base de datos.

<b>Pasos:</b>	<ul style="list-style-type: none"><li>• Iniciar sesión con un usuario registrado.</li><li>• Navegar al módulo donde se visualizan los permisos otorgados.</li><li>• Verificar los datos mostrados.</li></ul>
---------------	--

**Resultados Esperados:** Los usuarios deben visualizar sus roles y permisos asignados correctamente según los datos en la base de datos.



## HU006: Validar roles con base de datos

**Descripción:** Verificar la correcta asignación de roles y permisos a los usuarios mediante la base de datos.

<b>Pasos:</b>	<ul style="list-style-type: none"><li>• Acceder al sistema como administrador.</li><li>• Revisar y confirmar los permisos y roles asignados a los usuarios en la base de datos.</li></ul>
---------------	---

**Resultados Esperados:** El sistema debe reflejar los roles y permisos otorgados de manera precisa para cada usuario.

```

{
  "_id": "ObjectId('674627c1ff676bb85aff79a8')",
  "name": "prueba",
  "email": "prueba@gmail.com",
  "password": "e2a9109eMjGtN4pE9mzy3PwgsYRmDeLyHteVtoB7eGefyeCOMccQBMTNDq",
  "role": "user",
  "__v": 0
}

{
  "_id": "ObjectId('67462f03ff676bb85aff79ad')",
  "name": "PRUEBA",
  "email": "prueba@gmail.com",
  "password": "e2a9109eMjGtN4pE9mzy3PwgsYRmDeLyHteVtoB7eGefyeCOMccQBMTNDq",
  "role": "admin",
  "__v": 0
}

```

### SPRINT 3:

#### HU007: Crear rutinas personalizadas

**Descripción:** Validar que un administrador pueda crear rutinas personalizadas para los usuarios.

**Pasos:**

- Iniciar sesión como administrador.
- Acceder al módulo de creación de rutinas.
- Ingresar los detalles de la rutina personalizada.
- Guardar la rutina en el sistema.

**Resultados Esperados:** El administrador debe poder crear y guardar rutinas personalizadas con éxito.

**Dashboard Admin**

**Rutinas ejercicios**

**Suscribirse**

**Contraseña**

**Role**: Usuario

**Añadir**

### Listado de Usuarios

#	Nombre	Email	Acciones
1	prueba	prueba@gmail.com	Editar Eliminar Asignar Rutina Asignar Dieta
2	PRUEBA	prueba1@gmail.com	Editar Eliminar Asignar Rutina Asignar Dieta
3	prueba	administrador1@gmail.com	Editar Eliminar Asignar Rutina Asignar Dieta

### Perfil de prueba

**Abrir perfil**

**Lista de Usuarios**

#	Nombre	Email	Acciones
1	prueba	prueba@gmail.com	Editar Eliminar Asignar Rutina Asignar Dieta
2	PRUEBA	prueba1@gmail.com	Editar Eliminar Asignar Rutina Asignar Dieta
3	prueba	administrador1@gmail.com	Editar Eliminar Asignar Rutina Asignar Dieta

### Perfil de prueba

**Cerrar perfil**

**Lista de Usuarios**

#	Nombre	Email	Acciones
1	prueba	prueba@gmail.com	Editar Eliminar Asignar Rutina Asignar Dieta
2	PRUEBA	prueba1@gmail.com	Editar Eliminar Asignar Rutina Asignar Dieta
3	prueba	administrador1@gmail.com	Editar Eliminar Asignar Rutina Asignar Dieta

### Perfil de prueba

**Cerrar perfil**

### Asignar ejercicios

**SENADILLAS**

REALIZAR SENTADILLA SIN PESO EXTRA. 4 SERIES DE 12 REPETICIONES.

<https://www.youtube.com/watch?v=BjzczWeW4EY>

**Añadir ejercicio**

**Lista de Usuarios**

#	Nombre	Email	Acciones
1	prueba	prueba@gmail.com	Editar Eliminar Asignar Rutina Asignar Dieta
2	PRUEBA	prueba1@gmail.com	Editar Eliminar Asignar Rutina Asignar Dieta
3	prueba	administrador1@gmail.com	Editar Eliminar Asignar Rutina Asignar Dieta

### Perfil de prueba

**Cerrar perfil**

### Asignar ejercicios

**Nombre de ejercicio**

**Descripción (ej, tiempo, repeticiones)**

**Enlace de video (YouTube)**

**Añadir ejercicio**

**SENADILLAS**

REALIZAR SENTADILLA SIN PESO EXTRA. 4 SERIES DE 12 REPETICIONES.

**Editar Eliminar**

## HU008: Asignar rutinas a usuarios

**Descripción:** Comprobar que un administrador pueda asignar rutinas personalizadas a los usuarios registrados.

- |               |  |
|---------------|--|
| <b>Pasos:</b> | <ul style="list-style-type: none"><li>• Seleccionar una rutina creada previamente.</li><li>• Elegir un usuario registrado.</li><li>• Asignar la rutina al usuario y guardar los cambios.</li></ul> |
|---------------|--|

**Resultados Esperados:** La rutina debe aparecer correctamente en el perfil del usuario asignado.

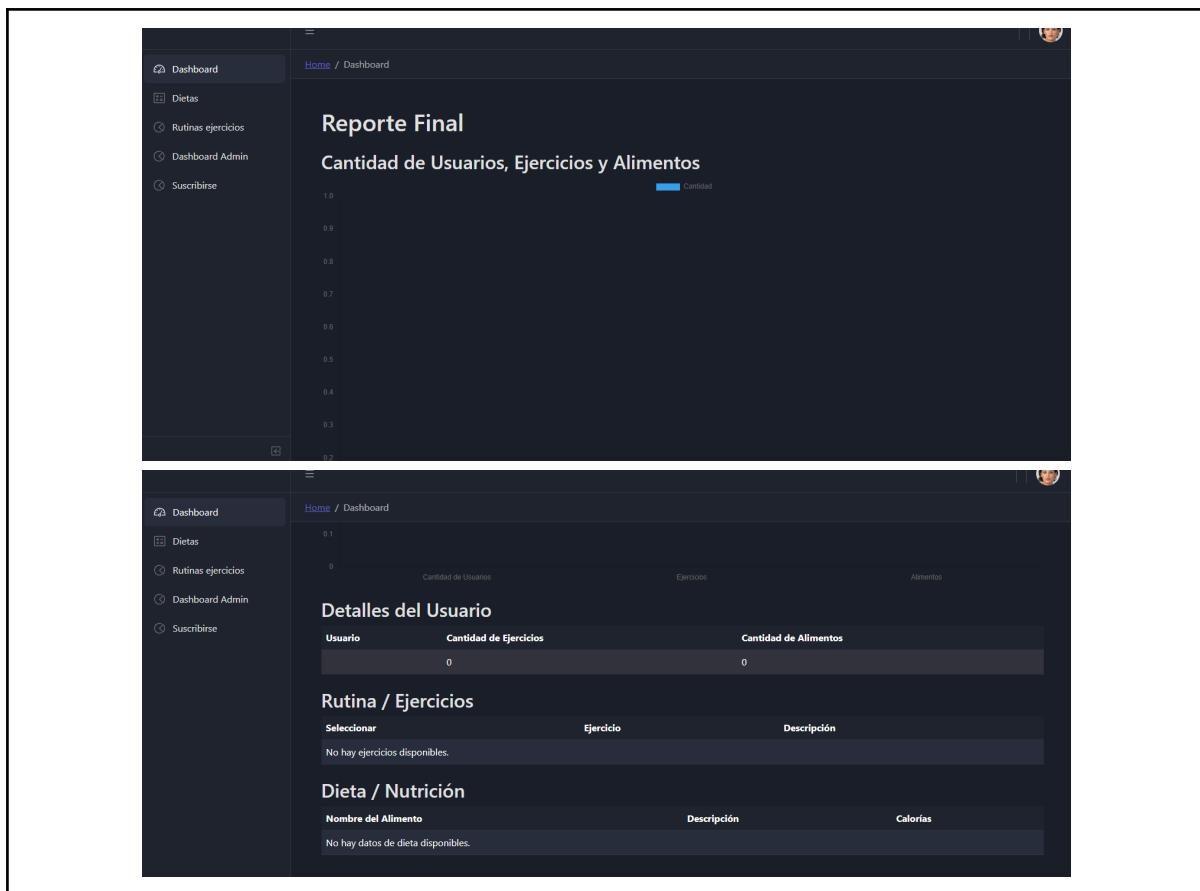
The image consists of two vertically stacked screenshots of a web-based fitness application. The top screenshot is titled 'AdminDashboard' and shows a list of three users: 'prueba' (email: prueba@gmail.com), 'PRUEBA' (email: prueba1@gmail.com), and 'prueba' (email: administrador1@gmail.com). Each user entry has four buttons: 'Editar' (yellow), 'Eliminar' (red), 'Asignar Rutina' (blue), and 'Asignar Dieta' (green). The bottom screenshot is titled 'Exercises' and shows a single exercise entry for 'SENTADILLAS'. The description reads: 'REALIZAR SENTADILLA SIN PESO EXTRA. 4 SERIES DE 12 REPETICIONES.' Below the description are two buttons: 'Editar' (yellow) and 'Eliminar' (red).

## HU009: Ver el dashboard con información actualizada

**Descripción:** Verificar que los usuarios puedan visualizar su progreso diario en el Dashboard.

- |               |   |
|---------------|---|
| <b>Pasos:</b> | <ul style="list-style-type: none"><li>• Iniciar sesión como usuario.</li><li>• Completar las actividades asignadas.</li><li>• Revisar el Dashboard para verificar los avances actualizados.</li></ul> |
|---------------|---|

**Resultados Esperados:** El Dashboard debe reflejar los avances actualizados según las actividades completadas por el usuario.



## SPRINT 4:

### HU010: Crear dietas personalizadas

**Descripción:** Confirmar que un administrador o nutricionista pueda crear dietas personalizadas para los usuarios.

<b>Pasos:</b>	<ul style="list-style-type: none"> <li>● Iniciar sesión con permisos de administrador.</li> <li>● Acceder al módulo de creación de dietas.</li> <li>● Diseñar una dieta personalizada y guardarla.</li> </ul>
---------------	---

**Resultados Esperados:** La dieta personalizada debe ser creada y registrada correctamente en el sistema.

The image contains two vertically stacked screenshots of a web application's admin dashboard.

**Screenshot 1 (Top):**

- Left Sidebar:** Includes links for Dashboard, Dietas, Rutinas ejercicios, Dashboard Admin, and Suscribirse.
- Header:** Shows "Home / AdminDashboard" and a "Añadir" button.
- Title:** "Lista de Usuarios".
- Table:** Displays three users with columns for #, Nombre, Email, and Acciones (Edit, Eliminar, Asignar Rutina, Asignar Dieta).

**Screenshot 2 (Bottom):**

- Left Sidebar:** Same as Screenshot 1.
- Header:** Shows "Home / AdminDashboard" and a "Añadir" button.
- Title:** "Lista de Usuarios".
- Table:** Displays three users with columns for #, Nombre, Email, and Acciones (Edit, Eliminar, Asignar Rutina, Asignar Dieta).
- Form:** "Asignar Dieta a prueba" with fields for PASAS (50) and Descripción (Comer por las mañanas), and buttons for "Asignar Dieta" and "Cancelar".
- Title:** "Dietas Asignadas".
- Table:** Displays assigned diets with columns for #, Nombre del Alimento, Calorías, Descripción, and Acciones.

## HU011: Asignar dietas a usuarios

**Descripción:** Validar que las dietas creadas puedan ser asignadas a los usuarios.

### Pasos:

- Seleccionar una dieta creada previamente.
- Elegir un usuario registrado.
- Asignar la dieta y guardar los cambios.

**Resultados Esperados:** El usuario debe poder visualizar la dieta asignada en su perfil.

The screenshot shows a browser window with the URL "localhost:3000/#/diets".

**Left Sidebar:** Includes links for Dashboard, Dietas, Rutinas ejercicios, Dashboard Admin, and Suscribirse.

**Header:** Shows "Home / Dietas".

**Title:** "Gestión de Dietas".

**Section:** "Lista de Dietas".

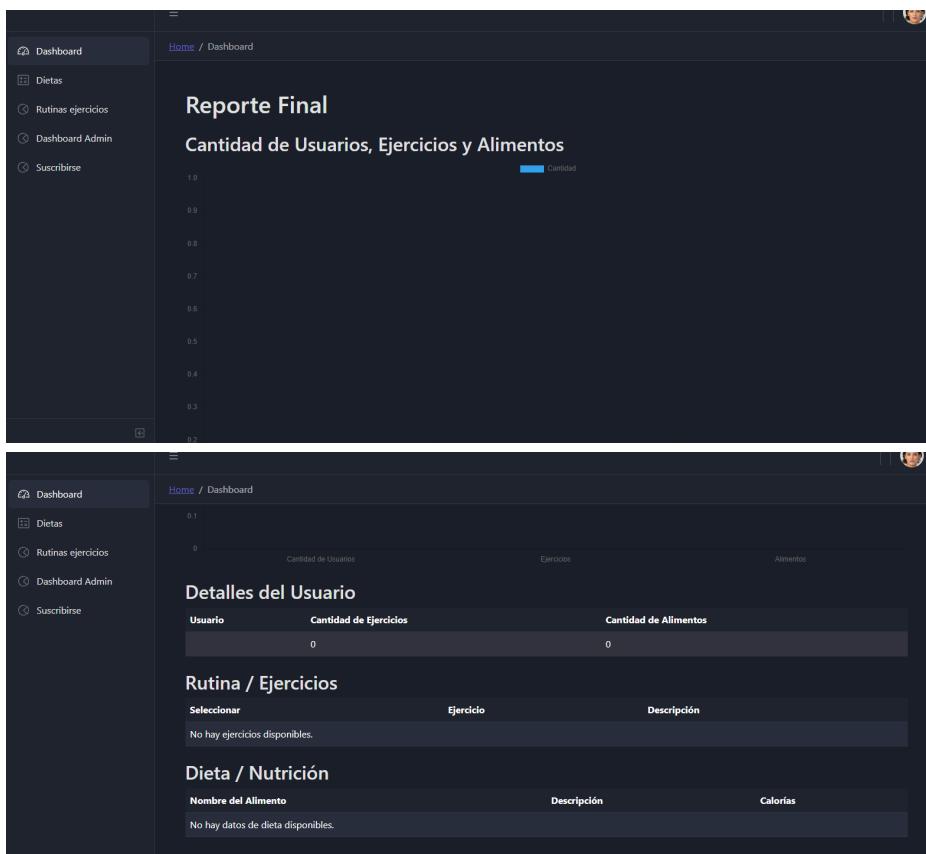
**Table:** Displays a header row with columns for #, Nombre del Alimento, Calorías, and Descripción.

## HU012: Ver dietas en el dashboard

**Descripción:** Verificar que los usuarios puedan observar sus dietas personalizadas en el Dashboard.

- |               |  |
|---------------|--|
| <b>Pasos:</b> | <ul style="list-style-type: none"><li>• Iniciar sesión como usuario.</li><li>• Navegar al Dashboard.</li><li>• Revisar los datos de la dieta asignada.</li></ul> |
|---------------|--|

**Resultados Esperados:** La dieta personalizada debe aparecer correctamente en el Dashboard y reflejar actualizaciones en tiempo real.



## CONCLUSIONES

A pesar de los desafíos enfrentados durante la etapa inicial de integración, el proyecto logró implementar exitosamente los módulos de generación y gestión de informes, marcando un avance significativo en la automatización del manejo de historias clínicas. Sin embargo, aún existen aspectos que necesitan ajustes para garantizar un funcionamiento más eficiente y fluido.

La funcionalidad de personalización de informes se ha desarrollado de manera básica, permitiendo a los usuarios finales modificar ciertos elementos. Aunque este es solo un primer paso, representa una base prometedora hacia una personalización más avanzada que se seguirá perfeccionando para atender mejor las necesidades específicas de los usuarios. Además, la generación de informes ha mostrado una notable mejora en velocidad y manejo eficiente de datos, reduciendo considerablemente los tiempos de espera para los usuarios.

En general, el desarrollo del sistema ha sentado las bases para una gestión más eficiente y automatizada de las rutinas y dietas en el desarrollo de los usuarios. Aunque los avances son significativos, se ha identificado la necesidad de continuar optimizando y actualizando el sistema para adaptarse a los avances tecnológicos y a las crecientes expectativas de los usuarios.

## RECOMENDACIONES

1. Programar reuniones periódicas para mantener al equipo informado sobre los avances y abordar cualquier inconveniente que surja durante el desarrollo.
2. Garantizar que toda la documentación esté clara y accesible, incluyendo los diseños del sistema y los manuales de usuario, para facilitar el progreso del proyecto.
3. Mantener una comunicación constante con los demás equipos involucrados en el proyecto para coordinar avances, ya que todo se integrará en un sistema único al finalizar.

4. Redactar historias de usuario detalladas que describan claramente cómo los diferentes usuarios interactúan con el sistema, asegurando una comprensión completa de los requerimientos.
5. Implementar metodologías ágiles, como Scrum o Kanban, para gestionar el desarrollo, promoviendo una respuesta rápida a los cambios y mejoras continuas en el proceso.

## **ANEXOS**

**Anexo 01. Manual Técnico**

[https://docs.google.com/document/d/1-gNe7f0AGeLy5bPQ\\_8yw7MQC3WPlzOIHuZwL00AsvwQ/edit?tab=t.0](https://docs.google.com/document/d/1-gNe7f0AGeLy5bPQ_8yw7MQC3WPlzOIHuZwL00AsvwQ/edit?tab=t.0)

**Anexo 02. Manual de Usuario**

[https://docs.google.com/document/d/1\\_udmm30a6R5iEQWZh42u1JZ7exMYGT2ht9uk3khi5OE/edit?tab=t.0](https://docs.google.com/document/d/1_udmm30a6R5iEQWZh42u1JZ7exMYGT2ht9uk3khi5OE/edit?tab=t.0)