UNIVERSIDAD CONTINENTAL

FACULTAD DE INGENIERÍA

ESCUELA ACADÉMICO PROFESIONAL DE INGENIERÍA DE SISTEMAS E INFORMÁTICA

PROYECTO

"Desarrollo de una aplicación de entrenamiento personal."

PRESENTADO POR:

APELLIDOS Y NOMBRES	CÓDIGO
Becerra Atencio Franz	47960727
Chahuin Miraval Xiomara Solange	74091952
Coronado Huamaní Edilson	72238029
Güere Casas Jose Manuel	72047656
Hilario Caso Hieyson Steven	77486405
Huertos Roque Jean Esteven	47375221
Mauricio Orihuela Isaac	72582650
Quispe Quispe Florentino Eduardo	47670934
Solis Torres Jose Eduardo	77326755
Torres de la cruz Jhon Henry	72882964
Torres Torres Angela Paola	75026225

ASESOR: Daniel Gamarra Moreno

HUANCAYO – PERÚ

2024

LISTA DE CONTENIDO

PRESENTADO POR: 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 10
1.4. Oportunidad Encontrada 10
1.5. Detalles del Proyecto 10
CAPÍTULO 2 11
ESTUDIO DE FACTIBILIDAD 11
2.1. Alternativas de Solución 11
2.2. Factibilidad Técnica 12
2.2.1. Infraestructura Requerida 12
2.2.2. Capacidades del Equipo de Desarrollo 12
2.3. Factibilidad Económica 12
2.3.1. Gastos generales 12
2.4. Factibilidad Operacional 12
2.4.1. Sistemas de ventas 12
CAPÍTULO 3 13
ANÁLISIS DE REQUERIMIENTOS 13
3.1. Metas del Sistema de Información 13
3.2. Requisitos del Sistema 13
3.2.1. Requerimientos funcionales 13
3.2.2. Requerimientos no funcionales 15
3.3. Identificación de Actores del Sistema 15
3.3.1. Usuario Registrado: Persona que crea una cuenta y utiliza el sistema para seguir rutinas de ejercicio. 15
3.3.2. Administrador: Persona encargada de gestionar el sistema, incluyendo la creación y actualización de rutinas. 15
CAPÍTULO 4 16
PLANIFICACIÓN DEL PROYECTO 16
4.1. Definición de Roles de Trabajo 16
4.1.1. Product owner 16
4.1.2. Scrum master 16
4.1.3. Team member 16
4.1.4. Tester 16
4.2. Product Backlog 16

4.3. Sprint Backlog 16
4.3.1. Sprint 1 16
4.3.2. Sprint 2 16
4.3.3. Sprint 3 16
4.3.4. Sprint 4 16
4.3.5. Sprint 5 17
4.4. Planificación de Sprints 17
4.4.1. Historias de usuario 17
4.4.2. Priorización de historias de usuario 17
4.5. Cronograma de Actividades 17
4.6. Gestión de Riesgos 17
CAPÍTULO 5 18
DISEÑO DEL SISTEMA DE INFORMACIÓN 18
5.1. Diseño de Diagramas UML 18
5.1.1. Diagramas de casos de uso 18
5.1.2. Diagramas de secuencia 18
5.1.3. Diagramas de colaboración 18
5.1.4. Diagramas de clases 18
5.2. Diseño de Base de Datos 18
5.2.1. Diseño conceptual (E/R) 18
5.2.2. Diseño lógico 18
5.2.3. Diseño físico 18
5.2.4. Modelado de base de datos 18
5.3. Diseño de Interfaces Básicas 18
5.3.1. Acceso login 19
5.3.2. Interfaz … 19
CAPÍTULO 6 20
CODIFICACIÓN DEL SOFTWARE 20
6.1. Desarrollo del Sprint 1 20
6.1.1. Sprint planning 20
6.1.2. Sprint backlog 20
6.1.3. Historias de usuarios 20
6.1.4. Taskboard 20
6.1.5. Daily scrum 20
6.1.6. Sprint review 20
6.1.7. Criterios de aceptación 20
6.1.8. Resultados del sprint 20
6.1.8.1. Evidencias. 20
6.1.8.2. Prueba de desarrollo. 20
6.1.8.3. …. 21
6.1.9. Sprint retrospective 21
6.2. Desarrollo del Sprint 2 21
6.2.1. Sprint planning 21

6.2.2. Sprint backlog 21
6.2.3. Historias de usuarios 21
6.2.4. Taskboard 21
6.2.5. Daily scrum 21
6.2.6. Sprint review 21
6.2.7. Criterios de aceptación 21
6.2.8. Resultados del sprint 21
6.2.8.1. Evidencias. 21
6.2.8.2. Prueba de desarrollo. 22
6.2.8.3. …. 22
6.2.9. Sprint retrospective 22
6.3. Desarrollo del Sprint 3 22
6.3.1. Sprint planning 22
6.3.2. Sprint backlog 22
6.3.3. Historias de usuarios 22
6.3.4. Taskboard 22
6.3.5. Daily scrum 22
6.3.6. Sprint review 22
6.3.7. Criterios de aceptación 22
6.3.8. Resultados del sprint 22
6.3.8.1. Evidencias. 23
6.3.8.2. Prueba de desarrollo. 23
6.3.8.3. …. 23
6.3.9. Sprint retrospective 23
6.4. Desarrollo del Sprint 4 23
6.4.1. Sprint planning 23
6.4.2. Sprint backlog 23
6.4.3. Historias de usuarios 23
6.4.4. Taskboard 23
6.4.5. Daily scrum 23
6.4.6. Sprint review 23
6.4.7. Criterios de aceptación 23
6.4.8. Resultados del sprint 24
6.4.8.1. Evidencias. 24
6.4.8.2. Prueba de desarrollo. 24
6.4.8.3. …. 24
6.4.9. Sprint retrospective 24
CAPÍTULO 7 25
PRUEBAS DE SOFTWARE 25
7.1. Plan de Pruebas 25
CONCLUSIONES 26
RECOMENDACIONES 27
ANEXOS 28

Anexo 01. Manual Técnico 29
Anexo 02. Manual de Usuario 30

CAPÍTULO 1
PLANTEAMIENTO DEL ESTUDIO

1.1. Aspectos Generales de la Empresa

Go Fitness Gym es un gimnasio privado ubicado en la ciudad de Huancayo, Perú. Fundado en 2010, se ha destacado como un referente en el sector fitness de la región central del país, ofreciendo servicios de alta calidad para promover la salud y el bienestar de sus clientes.

Entre los servicios que ofrece se encuentran:

- Entrenamiento personalizado.
- Clases grupales (zumba, yoga, spinning, pilates, entre otros).
- Evaluaciones físicas periódicas.
- Programas de rehabilitación física.
- Nutrición deportiva personalizada.

Con un equipo de profesionales y maquinaria de última generación, Go Fitness Gym busca ser un espacio integral para el desarrollo físico y mental de sus clientes.

1.1.1. Organigrama

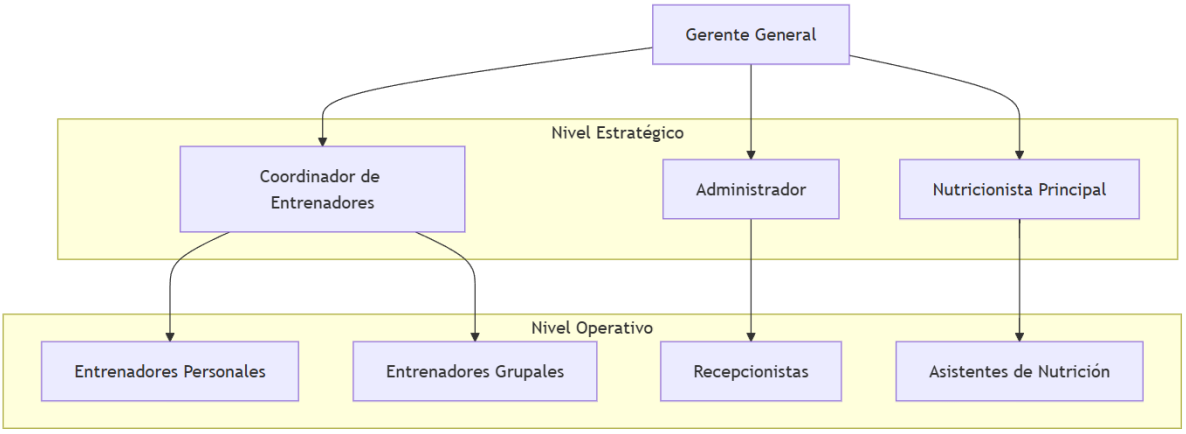


Figura 1. Organigrama

1.1.2. Misión y visión

Misión: Fomentar el bienestar integral de nuestros clientes a través de programas de entrenamiento personalizados, promoviendo un estilo de vida saludable y activo en la comunidad de Huancayo.

Visión: Convertirnos en el gimnasio líder de la región central del Perú, reconocidos por nuestra innovación, calidad de servicio y el impacto positivo en la vida de nuestros clientes.

1.2. Diagnóstico del Problema

El proyecto para desarrollar una aplicación de entrenamiento personal surge como respuesta a los siguientes desafíos identificados en la gestión y operación de Go Fitness Gym en Huancayo:

- **Gestión Manual de Planes de Entrenamiento:** Actualmente, los entrenadores realizan el seguimiento de rutinas y objetivos de forma manual, lo que dificulta una evaluación continua y precisa del progreso de los clientes.
- **Limitada Disponibilidad de Información para los Clientes:** Los usuarios no tienen acceso centralizado a información como su historial de ejercicios, métricas de progreso, y horarios de clases.
- **Procesos Operativos no Automatizados:** Las reservas de clases grupales y el registro de asistencia son gestionados manualmente, lo que genera demoras y problemas de disponibilidad.

1.3. Procesos de la Empresa

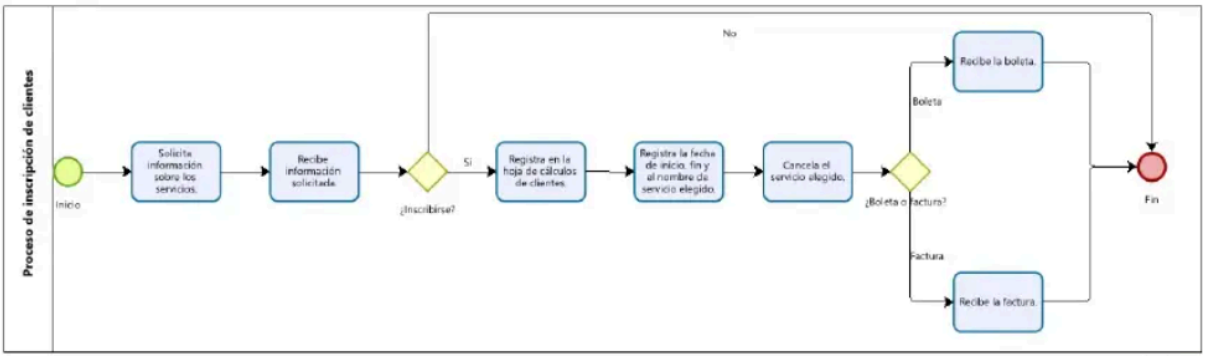


Figura 2. Procesos de la Empresa

1.4. Oportunidad Encontrada

Los desafíos actuales presentan una oportunidad única para transformar la gestión y experiencia del cliente mediante una solución tecnológica:

2. **Seguimiento Personalizado Automatizado:** Implementar una aplicación que permita a los entrenadores y clientes acceder fácilmente a rutinas personalizadas y métricas de progreso.
3. **Mejora de la Comunicación:** La aplicación permitirá una interacción directa y estructurada entre entrenadores y clientes, optimizando el tiempo y reduciendo malentendidos.
4. **Automatización de Reservas:** La digitalización del sistema de reservas para clases grupales agilizará la gestión de horarios, reduciendo errores y mejorando la satisfacción de los usuarios.
5. **Motivación y Retención:** Ofrecer herramientas para visualizar avances a través de gráficos y reportes incrementará la motivación de los clientes y su compromiso con sus objetivos.

5.1. Detalles del Proyecto

El proyecto consiste en diseñar e implementar un aplicativo web para Go Fitness Gym el cual será TrainUp, orientado a la gestión integral de entrenamientos personales y operativos del gimnasio.

Características Clave:

- Gestión personalizada de rutinas de entrenamiento.
- Monitoreo de métricas físicas y progreso.
- Sistema de reservas para clases grupales.

Fases del Proyecto:

1. **Análisis de Requisitos:** Identificación de necesidades específicas del gimnasio y sus clientes.
2. **Diseño:** Creación de prototipos de la interfaz de usuario y definición de funcionalidades.
3. **Desarrollo:** Implementación técnica de la aplicación.
4. **Pruebas:** Verificación y ajuste del sistema antes de su lanzamiento.
5. **Implementación:** Despliegue y activación de la aplicación en el gimnasio.
6. **Capacitación:** Entrenamiento a entrenadores y personal en el uso de la herramienta.
7. **Mantenimiento:** Actualizaciones y soporte técnico continuo.

Beneficios:

- Incremento en la satisfacción y retención de los clientes.
- Mayor eficiencia en la gestión de recursos y servicios.
- Mejora en la comunicación interna y externa.

- Posicionamiento de Go Fitness Gym como un referente en innovación tecnológica en el sector fitness.

CAPÍTULO 2
ESTUDIO DE FACTIBILIDAD

2.1. Alternativas de Solución

En el desarrollo del sistema propuesto se consideran las siguientes alternativas de solución:

Propuesta 01: **Uso de Tecnologías Tradicionales** Se planteó desarrollar la aplicación utilizando tecnologías más convencionales sin contenedorización. Este enfoque implica un desarrollo más estático y dependiente de servidores físicos con configuraciones específicas. Sin embargo, esta alternativa carece de escalabilidad y flexibilidad, lo que podría limitar la capacidad de personalización y el crecimiento futuro de la aplicación.

Propuesta 02: **Desarrollo con Stack Moderno MERN y Contenedorización con Docker** Esta alternativa contempla el uso de tecnologías modernas como MongoDB, Express.js, React.js y Node.js (stack MERN), complementadas con Docker para la contenedorización. Este enfoque permite un desarrollo modular, facilita el despliegue en distintos entornos y asegura una experiencia fluida para los usuarios.

Propuesto 03: **Adquisición de un Software Preexistente** Finalmente, se analizó la posibilidad de adquirir un sistema ya desarrollado. Aunque esta opción implica menores tiempos de implementación, no se ajusta a las necesidades específicas del proyecto , no permite personalizar funcionalidades específicas ni optimizar los datos de progreso de los usuarios.

Tras el análisis, se seleccionó la segunda alternativa (stack MERN + Docker) por su flexibilidad, escalabilidad y capacidad para adaptarse a las necesidades específicas de la aplicación de entrenamiento personal. Este enfoque garantiza una experiencia optimizada para los usuarios, con funcionalidades personalizables y un despliegue eficiente en diversos entornos.

2.2. Factibilidad Técnica

El análisis de la factibilidad técnica incluye los siguientes puntos

2.2.1. Infraestructura Requerida

Hardware: Se necesitarán servidores confiables y de alta capacidad para alojar tanto la base de datos como la lógica de la aplicación. Estos deben estar preparados para soportar múltiples conexiones simultáneas de los usuarios y manejar eficientemente grandes cantidades de información, como datos de progreso y configuraciones personalizadas.

El equipo de desarrollo requiere computadoras con especificaciones adecuadas para trabajar con tecnologías como Docker, React.js y Node.js. Esto asegura que el entorno de desarrollo sea estable y compatible con las herramientas utilizadas.

Software:

Sistema Operativo-Windows 10 o superior, Linux Ubuntu 20.04+

Navegador Web-Google Chrome (recomendado), Brave

Herramientas de Desarrollo- Visual Studio Code, Docker, MongoDB Compass

Lenguajes y Frameworks-Node.js, React.js, Express.js

Herramientas: Git Flow para el control de versiones y metodologías ágiles para el desarrollo colaborativo.

2.2.2. Capacidades del Equipo de Desarrollo

El equipo cuenta con experiencia previa en desarrollo web y aplicaciones móviles utilizando tecnologías modernas. No obstante, se identificó la necesidad de capacitación en contenedorización con Docker, la cual ya está planificada.

2.2.3. Mitigación de Riesgos Técnicos

Riesgo: Dificultades en la integración del stack MERN con Docker.

Acción: Realizar pruebas de integración frecuentes y garantizar sesiones de capacitación específicas.

2.2.4. Cronograma de Desarrollo

El proyecto está diseñado para completarse en un plazo de cuatro meses, con etapas bien definidas para el diseño, desarrollo, pruebas y despliegue.

2.3. Factibilidad Económica

La aplicación permitirá a los usuarios gestionar rutinas personalizadas, registrar su progreso y recibir recomendaciones adaptadas a sus necesidades. Esto mejorará su experiencia de usuario y posicionará la plataforma como una solución integral en el mercado fitness.

2.3.1. Gastos generales

Se contemplan costos adicionales de mantenimiento y actualizaciones, estimados en S/5,000 anuales, para garantizar la calidad y funcionalidad continua de la aplicación**.**

Desarrollo de la Aplicación: S/40,000.

Licencias de Software: S/5,000.

Infraestructura Adicional: S/5,000.

Total estimado: **S/50,000**.

2.4. Factibilidad Operacional

El proyecto de la aplicación de entrenamiento personal TrainUp busca resolver desafíos en la gestión y personalización de rutinas de entrenamiento, motivación y seguimiento del progreso de los usuarios. Para abordar estos retos, se propone la implementación de un software integral diseñado para optimizar la experiencia del usuario en actividLa solución propuesta combina funciones clave como la creación de rutinas personalizadas, registro y análisis del progreso, y recomendaciones basadas en inteligencia artificial. Su implementación promete transformar la manera en que los

usuarios alcanzan sus objetivos de entrenamiento mediante una plataforma unificada y accesible.

El software será desarrollado con un enfoque en la calidad y experiencia del usuario, destacando una interfaz amigable y accesible que facilite su adopción y uso. Los beneficios previstos incluyen:ades físicas.

Automatización de procesos clave, como el diseño de planes de entrenamiento personalizados y el envío de notificaciones, ahorrando tiempo y reduciendo el margen de error.

Acceso centralizado a información detallada de rutinas, progreso y estadísticas, lo que permitirá a los usuarios monitorear sus avances de manera intuitiva.

Herramientas avanzadas para entrenadores, que facilitarán la supervisión de clientes y la retroalimentación directa basada en datos precisos.

Motivación constante a través de recordatorios, desafíos semanales y mensajes personalizados.

2.4.1. Plan de Capacitación

Para garantizar la adopción exitosa del sistema, se ha diseñado un plan integral de formación para los usuarios y entrenadores:

Capacitación del Equipo Inicial:

Un grupo selecto de entrenadores y administradores recibirán formación intensiva en el uso de la aplicación, incluyendo la personalización de rutinas y el análisis de progreso.

Este equipo será responsable de liderar la implementación en centros de entrenamiento.

Efecto Multiplicador:

Los entrenadores capacitados serán responsables de guiar a sus colegas y clientes en el uso del sistema, asegurando una adopción uniforme en la comunidad de usuarios.

Manuales y Recursos de Apoyo:

Se desarrollará un Manual de Usuario detallado, acompañado de tutoriales interactivos, que permitirá a los usuarios aprender a manejar las funcionalidades clave de la aplicación.

Feedback Continuo:

Durante las primeras etapas de uso, se habilitarán canales de soporte para resolver dudas y recibir sugerencias de los usuarios, mejorando la funcionalidad y usabilidad del sistema.

Viabilidad Operacional

La implementación del software cuenta con el respaldo de los interesados clave, como entrenadores, usuarios finales y patrocinadores del proyecto. La infraestructura tecnológica planificada, junto con la estrategia de capacitación, asegura que el sistema sea integrado de manera efectiva en el flujo de trabajo diario.

Gracias a su diseño adaptable, el software no solo responde a las necesidades actuales, sino que también tiene la capacidad de evolucionar con el tiempo, incorporando nuevas funciones según los comentarios de los usuarios y avances tecnológicos.

CAPÍTULO 3
ANÁLISIS DE REQUERIMIENTOS

3.1. Metas del Sistema de Información

Crear y gestionar rutinas de entrenamiento: El sistema debe permitir la creación, edición y seguimiento de rutinas de ejercicio.

Visualizar y analizar estadísticas de progreso: El sistema debe registrar, procesar y presentar datos de rendimiento del usuario.

Proveer recomendaciones personalizadas de ejercicio: El sistema debe generar sugerencias de entrenamiento basadas en el historial y objetivos del usuario.

3.2. Requisitos del Sistema

3.2.1. Requerimientos funcionales

Código RF	Nombre	Descripción
RF1	Registro de Usuario	Permitir a los usuarios crear una cuenta con nombre, correo electrónico y contraseña.
RF2	Inicio de Sesión	Autenticar a los usuarios registrados mediante correo electrónico y contraseña.
RF3	Selección de Nivel de Dificultad	Permitir a los usuarios seleccionar su nivel de dificultad (principiante, intermedio, avanzado).
RF4	Generación de Rutinas Básicas	Ofrecer rutinas de entrenamiento predeterminadas basadas en el nivel de dificultad seleccionado.

RF5	Menú de Rutinas	Proveer un menú donde los usuarios puedan seleccionar y seguir una rutina diaria.
RF6	Mostrar detalles de la Rutina	Permitir a los usuarios hacer clic en una rutina para ver más detalles. Mostrar una descripción detallada de cada ejercicio, incluyendo repeticiones, series y tiempo de descanso.
RF7	Personalización de Rutinas	Generar rutinas personalizadas en función de los objetivos y el nivel de dificultad del usuario.
RF8	Gestión de Preferencias y Programación	Permitir a los usuarios ajustar sus preferencias de notificaciones, horario, etc. También permitirá programar rutinas, incluyendo la posibilidad de establecer la hora del entrenamiento.
RF9	Menú de Rutinas Personalizadas	Proveer un menú donde los usuarios puedan seleccionar y seguir sus rutinas personalizadas creadas anteriormente.
RF10	Notificaciones	Enviar recordatorios y notificaciones basadas en las preferencias del usuario, como la hora de las rutinas y los objetivos establecidos.
RF11	Seguimiento de Progreso	Registrar y mostrar únicamente las calorías gastadas y el tiempo total de entrenamiento.

RF13	Creación de Calendario Semanal	Permitir a los usuarios crear un calendario semanal que incluya las rutinas personalizadas creadas anteriormente, con la posibilidad de programarlas para cada día de la semana.

3.2.2. Requerimientos no funcionales

Categoría	Descripción
Seguridad	Asegurar que los datos de los usuarios estén protegidos mediante encriptación y autenticación segura.
Usabilidad	Diseñar una interfaz intuitiva y fácil de usar.
Rendimiento	Garantizar tiempos de respuesta rápidos y una experiencia de usuario fluida.
Compatibilidad	Asegurar que el sistema sea compatible con múltiples dispositivos y navegadores.

3.3. Identificación de Actores del Sistema

3.3.1. **Usuario Registrado: Persona que crea una cuenta y utiliza el sistema para seguir rutinas de ejercicio.**

3.3.2. **Administrador: Persona encargada de gestionar el sistema, incluyendo la creación y actualización de rutinas.**

CAPÍTULO 4
PLANIFICACIÓN DEL PROYECTO

4.1. Definición de Roles de Trabajo

Tabla 1. Definición de Roles de Trabajo

Nombre	Puesto	Tel	Rol en el proyecto	Tipo de Relación y Seguimiento que debe darse con este stakeholder
Becerra Atencio Franz	Documentación	960371680	Cliente / Product Owner	Definición de Requisitos, Validación de Funcionalidades
Chahuin Miraval Xiomara Solange	Desarrollador	940774864	Desarrollador Backend	Desarrollo del Backend
Hilario Caso Hieyson Steven	Developer	935533845	Desarrolladora Frontend	Desarrollo del Frontend
Torres de la cruz Jhon Henry	Desarrollador	963256980	Ingeniero DevOps	Gestión de Infraestructura y Despliegue
Torres Torres Angela Paola	Tester	982212165	Especialista en Datos	Diseño de Base de Datos y Análisis de Datos
Quispe Quispe Florentino Eduardo	Desarrollador	918155661	Diseñador UX/UI	Diseño de UX/UI y Prototipos
Mauricio Orihuela Isaac	Scrum master	977151240	Analista de Requerimientos	Recolección y Documentación de Requisitos
Huertos Roque Jean Esteven	Documentación	977818351	Tester / QA	Pruebas y Garantía de Calidad
Güere Casas Jose Manuel	Diseñador	921469387	Especialista en Seguridad	Seguridad de la Aplicación
Solis Torres Jose Eduardo	Desarrollador	925486524	Coordinador de Proyecto	Gestión del Proyecto
Coronado Huamaní Edilson	Tester	968029590	Entrenadora Personal	Validación de Recomendaciones y Contenido Especializado

4.1.1. Product owner

Tabla 1. Product owner

INTEGRANTES	Rol
Becerra Atencio Franz	**Product Owner**

4.1.2. Scrum master

Tabla 1. Scrum master

INTEGRANTES	Rol
Torres Torres Angela Paola	**Scrum master**

4.1.3. Team member

Tabla 1. Team member

INTEGRANTES	Rol
Chahuin Miraval Xiomara Solange	**Equipo de desarrollo (base de datos)**
Coronado Huamaní Edilson	**Equipo de desarrollo (base de datos)**
Güere Casas Jose Manuel	**Equipo de desarrollo (backend)**
Hilario Caso Hieyson Steven	**Equipo de desarrollo (backend)**
Huertos Roque Jean Esteven	**Equipo de desarrollo (backend)**
Mauricio Orihuela Isaac	**Equipo de desarrollo (frontend)**
Quispe Quispe Florentino Eduardo	**Equipo de desarrollo (frontend)**
Solis Torres Jose Eduardo	**Equipo de desarrollo (frontend)**
Torres de la cruz Jhon Henry	**Equipo de desarrollo (frontend)**
Becerra Atencio Franz	**Equipo de desarrollo (backend)**

4.1.4. Tester

Tabla 1. Tester

INTEGRANTES	Rol
Huertos Roque Jean Esteven	**Tester**

4.2. Product Backlog

Backlog Detallado del Producto

Figura 3. Product Backlog

Backlog Detallado del Producto

Nombre del Proyecto: Desarrollo de aplicativo de entrenamiento personal

Dueño del Producto

HISTORIA DE USUARIO				TAREA		OTROS DATOS DE LA HISTORIA DE USUARIO O TAREA					
ID Historia de Usuario	Como (Rol)	Deseo...	Para...	ID Tarea	Descrición de la Tarea	Criterios de Aceptación	Prioridad	Estimación	Dependencias	Sprint	Estado
1	Usuario	Crear una cuenta con nombre, correo electrónico y contraseña	Comenzar a usar la aplicación	1,1	Crear una cuenta	Acceder a las funcionalidades de la aplicación	1	5 dias	Ninguna	1	Terminado
2	Usuario registrado	Iniciar sesión mediante correo electrónico y contraseña	Continuar usando la aplicación con mi cuenta	1,2	Iniciar sesión	Acceder a mi cuenta y guardar mi progreso	1	6 dias	1,1	1	Terminado
3	Usuario	Seleccionar mi nivel de dificultad (principiante, intermedio, avanzado)	Realizar rutinas adecuadas a mi nivel	1,3	Seleccionar nivel de dificultad	Recibir rutinas acordes a mi nivel	1	5 dias	1,2	1	Terminado
4	Usuario	Recibir rutinas predeterminadas basadas en mi nivel de dificultad	Entrenar de forma estructurada	1,4	Obtener rutinas de entrenamiento	Seguir un plan de entrenamiento diario	1	3 dias	1,3	1	Terminado
5	Usuario	Acceder a un menú con las rutinas diarias	Realizar la rutina elegida de manera sencilla	1,5	Acceder al menú de rutinas	Seleccionar la rutina diaria que quiero seguir	1	3 dias	1,4	1	Terminado
6	Usuario	Personalizar mis rutinas según objetivos (bajar de peso, tonificar, etc.)	Tener un plan adaptado a mis necesidades	1,6	Personalizar mis rutinas	Alcanzar objetivos específicos de entrenamiento	1	5 dias	1,5	1	Terminado
7	Usuario	Seguir las rutinas personalizadas	Alcanzar mis objetivos de manera eficaz	1,7	Seguir mis rutinas personalizadas	Cumplir con mi plan de entrenamiento	2	5 dias	1,6	2	Terminado
8	Usuario	Ajustar preferencias de notificaciones, horarios, etc.	Tener un plan organizado y recibir recordatorios	1,8	Configurar mis preferencias	Recibir notificaciones y recordatorios personalizados	2	6 dias	1,7	2	Terminado
10	Usuario	Acceder a un menú con mis rutinas personalizadas	Tener acceso rápido a mi plan de entrenamiento	1,9	Acceder al menú de rutinas personalizadas	Seleccionar la rutina diaria personalizada que quiero seguir	2	5 dias	1,8	2	Terminado
11	Usuario	Recibir notificaciones sobre mis rutinas y objetivos	Mantenerme motivado y organizado	1,10	Recibir notificaciones de mi plan	Recordar los entrenamientos programados	2	3 dias	1,9	2	Terminado
12	Usuario	Registrar y visualizar calorías gastadas y tiempo de entrenamiento	Monitorizar mi progreso de manera efectiva	1,11	Seguir mi progreso	Ver mis avances en el tiempo	2	4 dias	1,10	2	Terminado
13	Usuario	Crear un calendario con rutinas personalizadas y programarlas	Mantener un plan de entrenamiento organizado y seguirlo semanalmente	1,12	Crear un calendario semanal	Planificar mis rutinas para la semana	1	5 dias	1,11	2	Terminado
14	Administrador	Generar retos semanales	Definir retos segun progreso	2,1	Administrar retos semanales de entrenamiento	Llevar el control de rendiminto de los participantes	1	6 dias	Ninguna	3	Terminado
15	Administrador	Recopilar y mostrar puntajes	Evaluar el puntaje de los retos semanales	2,2	Visualizar panel de retos	Ver puntajes de todos los retos	1	5 dias	2,1	3	Terminado
16	Administrador	Asegurar la visibilidad y relevancia diaria de los mesajes a los participantes	Los mensajes de entrenamiento se destaquen diariamente	2,3	Mostrar mensajes motivacionales	Los mensajes aparezcan de manera destacada diariamente	1	5 dias	2,2	3	Terminado
17	Administrador	Acceder y controlar el sistema donde se controlan todos los datos	Tener acceso para hacer modificaciones de acuerdo a las solicitudes recibidas	2,4	Gestionar datos de todos los participantes	Modicicar y eliminar los datos de los participantes	1	5 dias	2,3	3	Terminado
18	Administrador	Exportar los datos de los usuarios y rutinas	Tener informes detallados y análisis externos	3,1	Generar reportes descargables en formato CSV o PDF	Los reportes deben contener datos filtrables por fecha y categoría	2	5 dias	2,4	3	Terminado
19	Administrador	Gestionar comentarios y feedback de los usuarios	Mejorar la experiencia del usuario y el contenido de la aplicación	3,2	Visualizar y moderar comentarios enviados por los usuarios	Debe existir un panel para moderar el contenido de forma sencilla	2	6 dias	3,1	4	Terminado
20	Administrador	Crear y gestionar múltiples administradores	Delegar tareas administrativas a otros usuarios autorizados	3,3	Definir roles y permisos para diferentes administradores	El sistema debe permitir asignar diferentes niveles de acceso a cada administrador	1	4 dias	3,2	4	Terminado

4.3. Sprint Backlog

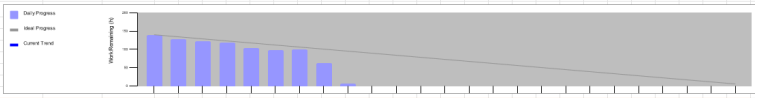
Backlog del Sprint.xlsx

4.3.1. Sprint 1

Figura 4. Backlog del Sprint 1

Nombre del Proyecto:	Desarrollo de una aplicación de entrenamiento personal
Dueño del Proyecto	Grupo 205
Gerente del Proyecto:	Torres Torres Angela Paola

Sprint 1 Backlog



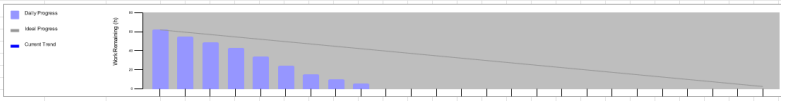
	Duración del Sprint	25				Esfuerzo	Faltante en los dias siguientes...																																		
	Tendencia calculada en los últimos	14	Dias		Totales	140	140	128	122	118	104	99	100	63	6																										
ID Tarea	Tarea	ID Historia	Responsable		Estado	Est.	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25										
1.1	Planificación del Proyecto	0	Franz		Terminado	15	15	13	11	9	7	5	3	2	1	0																									
1.2	Asignación de Roles y Recursos	0	Angela		Terminado	17	17	15	14	13	11	8	5	3	2	0																									
1.3	Criterios de Aceptación	0	Jean		Terminado	12	12	10	9	8	6	4	3	2	1	0																									
1.4	Documentación y Roadmap	0	Jose		Terminado	18	18	17	15	13	10	7	4	3	2	0																									
1.5	Configuración Inicial del Proyecto	0	Edison		Terminado	15	15	14	15	14	15	15	15	17																											
1.6	Registro de Usuario	HU01	Xiomara		Terminado	17	17	14	15	14	16	17	18	17																											
1.7	Inicio de Sesión	HU02	Isaac		Terminado	12	12	14	14	17	14	14	18	4																											
1.8	Selección de Nivel de Dificultad	HU03	Heison		Terminado	18	18	15	16	18	13	16	17	8																											
1.9	Generación de Rutinas Básicas	HU04	Jean		Terminado	16	16	16	13	12	12	13	17	7																											

4.3.2. Sprint 2

Figura 5. Backlog del Sprint 2

Nombre del Proyecto:	Desarrollo de una aplicación de entrenamiento personal
Dueño del Proyecto	Grupo 205
Gerente del Proyecto:	Torres Torres Angela Paola

Sprint 2 Backlog



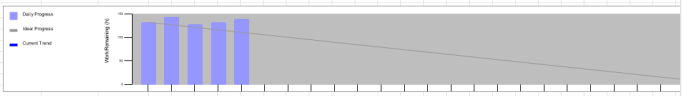
	Duración del Sprint	25				Esfuerzo	Faltante en los dias siguientes...																																		
	Tendencia calculada en los últimos	14	Dias		Totales	62	62	55	49	43	34	24	15	10	6																										
ID Tarea	Tarea	ID Historia	Responsable		Estado	Est.	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25										
2	Menú de Rutinas	HU05	Franz		Terminado	15	15	13	11	9	7	5	3	2	1	0																									
2.1	Personalización de Rutinas	HU06	Angela		Terminado	17	17	15	14	13	11	8	5	3	2	0																									
2.2	Gestión de Preferencias y Programación	HU07	Jean		Terminado	12	12	10	9	8	6	4	3	2	1	0																									
2.3	Menú de Rutinas Personalizadas	HU08	Jose		Terminado	18	18	17	15	13	10	7	4	3	2	0																									

4.3.3. Sprint 3

Figura 6. Backlog del Sprint 3

Nombre del Proyecto:	Desarrollo de una aplicación de entrenamiento personal
Dueño del Proyecto	Grupo 205
Gerente del Proyecto:	Torres Torres Angela Paola

Sprint 3 Backlog



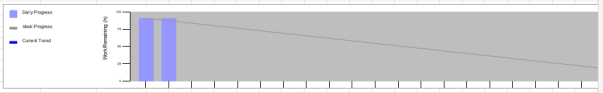
	Duración del Sprint	25				Esfuerzo	Faltante en los dias siguientes...																																		
	Tendencia calculada en los últimos	14	Dias		Totales	132	132	144	128	133	139																														
ID Tarea	Tarea	ID Historia	Responsable		Estado	Est.	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23												
3	Notificaciones	HU09	Franz		Terminado	15	15	19	18	17	15																														
3.1	Seguimiento de Progreso	HU10	Angela		Terminado	17	17	16	7	16	14																														
3.2	Creación de Calendario Semanal	HU11	Jean		Terminado	12	12	13	15	16	13																														
3.3	Implementar API para obtener rutinas según el nivel de dificultad seleccionado.	HU08	Jose		Terminado	18	18	14	15	12	15																														
3.4	Crear una pantalla para que el usuario seleccione su nivel de dificultad.	HU08	Edison		Terminado	15	15	16	15	17	17																														
3.5	Conectar el frontend con la API para la selección de nivel de dificultad.	HU08	Xiomara		Terminado	17	17	19	15	13	15																														
3.6	Implementar la visualización de rutinas básicas según el nivel de dificultad seleccionado.	HU09	Edison		Terminado	12	12	8	13	12	17																														
3.7	Mejora de Perfil y Autenticación	HU09	Jose		Terminado	12	12	20	17	12	18																														
3.8	Creacion de perfil de administrador	HU19	Eduardo		Terminado	14	14	19	13	18	15																														

4.3.4. Sprint 4

Figura 7. Backlog del Sprint 4

Nombre del Proyecto:	Desarrollo de una aplicación de entrenamiento personal
Dueño del Proyecto	Grupo 205
Gerente del Proyecto:	Torres Torres Angela Paola

Sprint 4 Backlog



	Duración del Sprint					25				Esfuerzo	Faltante en los dias siguientes...																								
	Tendencia calculada en los últimos					14	Dias		Totales	91	91	91																							
ID Tarea		Tarea				ID Historia	Responsable		Estado	Est.	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20					
	Administrar Retos Semanales					HU11	Franz		Terminado	15	15	15																							
	Visualizar Panel de Retos					HU12	Angela		Terminado	17	17	17																							
	Mostrar Mensajes Motivacionales					HU16	Jean		Terminado	12	12	12																							
	Gestionar Datos de los Participantes					HU18	Jose		Terminado	18	18	18																							
	Visualizar y Moderar Comentarios					HU20	Xiomara		Terminado	17	17	17																							
	Definir Roles y Permisos					HU20	Heison		Terminado	12	12	12																							

4.4. Planificación de Sprints

4.4.1. Historias de usuario

Backlog del Producto.xlsx

Figura 9. historias de usuario

Backlog del Producto

	Desarrollo de aplicativo de entrenamiento personal
Dueño del Producto	GRUPO 205

EPICA				HISTORIA DE USUARIO				OTROS DATOS DE LA EPICA O HISTORIA DE USUARIO					
ID Epica	Como (Rol)	Deseo...	Para...	ID Historia de Usuario	Como (Rol)...	Deseo....	Para....	Criterios de Aceptación	Prioridad	Estimación	Dependencias	Sprint	Estado
				HU01	Administrador	Deseo poder generar retos semanales	Administar retos semanales y los clientes puedan competir por lograrlos	Debe poder exportarse a pdf y xlsx Debe poderse filtrar por codigo de cliente	1	25		1	Por Hacer
				HU02	Administrador	Deseo visualizar panel de retos	Visualizar el cliente y los puntajes acumulados cada mes	Debe poder exportarse a pdf y xlsx Debe poderse filtrar por codigo de cliente	1	25	HU01	1	Por Hacer
				HU03	Administrador	Deseo poder mostrar mensajes motivacionales diariamente	Motivar a los clientes y mejorar su estado de animo	Debe permitir que los mensajes aparezcan de forma destacada Debe permitir perzonalizar los mensajes	1	25		2	Por Hacer
				HU04	Administrador	Deseo poder gestionar los datos de todos los clientes	Gestionar y llevar un control del numero de clientes	Debe permitir acceder y visualizar los datos Debe permitir modificar y eliminiar datos del cualquier cliente	1	25		2	Por Hacer
EPIC02	Entrenador	Poder generar rutinas personalizadas de entrenamiento.	Para mejorar la experiencia del usuario										
				HU05	Entrenador	Deseo actualizar rutinas basicas	Actualizar rutinas basicas y los usuarios puedan completarlas	Debe permitir acceder y visualizar los datos Debe permitir modificar y eliminiar datos del cualquier cliente	1	25		3	Por Hacer
				HU06	Entrenador	Deseo actualizar funcion de objetivos y una rutina diaria.	Agregar funiones y objeticos para que el usuario puedas cumplirlas	Debe permitir acceder y visualizar los datos Debe permitir modificar y eliminiar datos del cualquier cliente	1	25	HU05	3	Por Hacer
				HU07	Usuario	Deseo permitir que el sistema debe permitir ingresar u	Agregar el iniciar sesion y ingresar datos de usuario	Debe el sistema debe permitir ingresar un nombre, correo electrónico y contraseña.	2	20		1	Por Hacer
				HU08	Usuario	Deseo que se muestre un mensaje de confirmación de registro exitoso.	Visualizar mostrar un mensaje de confirmacion de registro exitoso	Debe mostrar un mensaje de confirmación de registro exitoso.	1	25	HU06	1	Por Hacer
EPIC04	Usuario	Deseo iniciar sesión en la aplicación	Para acceder a mis entrenamientos y configuraciones personales	HU09	USUARIO	Deseo que el sistema debe permitir ingresar un correo electrónico y contraseña válidos.	Visualizar un mensaje de contraseña y correo validos	Debe permitir ingresar el correo electronico y contraseñas validos	2	25	HU08	1	Por Hacer
				HU10	USUARIO	Deseo que redirigir al usuario a su panel principal tras una autenticación exitosa.	Visualizar panel de autentificacion	Debe auntenficar el usuario y contraseña	1	25	HU09	1	Por Hacer
				HU11	USUARIO	Deseo que el sistema debe mostrar un mensaje de er	Visualizar un mensaje de erros si las credenciales son incorrectas	Debe el sistema mostrar un mensaje de erros si las credenciales son incorrectas	1	20	HU10		Por Hacer

4.4.2. Priorización de historias de usuario

Backlog Detallado del Producto

Figura 10. Priorización de historias de usuario

Backlog Detallado del Producto

Nombre del Proyecto:	Desarrollo de aplicativo de entrenamiento personal
Dueño del Producto	

HISTORIA DE USUARIO				TAREA		OTROS DATOS DE LA HISTORIA DE USUARIO O TAREA					
ID Historia de Usuario	Como (Rol)	Deseo...	Para...	ID Tarea	Descrición de la Tarea	Criterios de Aceptación	Prioridad	Estimación	Dependencias	Sprint	Estado
1	Usuario	Crear una cuenta con nombre, correo electrónico y contraseña	Comenzar a usar la aplicación	1,1	Crear una cuenta	Acceder a las funcionalidades de la aplicación	1	5 dias	Ninguna	1	Terminado
2	Usuario registrado	Iniciar sesión mediante correo electrónico y contraseña	Continuar usando la aplicación con mi cuenta	1,2	Iniciar sesión	Acceder a mi cuenta y guardar mi progreso	1	6 dias	1,1	1	Terminado
3	Usuario	Seleccionar mi nivel de dificultad (principiante, intermedio, avanzado)	Realizar rutinas adecuadas a mi nivel	1,3	Seleccionar nivel de dificultad	Recibir rutinas acordes a mi nivel	1	5 dias	1,2	1	Terminado
4	Usuario	Recibir rutinas predeterminadas basadas en mi nivel de dificultad	Entrenar de forma estructurada	1,4	Obtener rutinas de entrenamiento	Seguir un plan de entrenamiento diario	1	3 dias	1,3	1	Terminado
5	Usuario	Acceder a un menú con las rutinas diarias	Realizar la rutina elegida de manera sencilla	1,5	Acceder al menú de rutinas	Seleccionar la rutina diaria que quiero seguir	1	3 dias	1,4	1	Terminado
6	Usuario	Personalizar mis rutinas según objetivos (bajar de peso, tonificar, etc.)	Tener un plan adaptado a mis necesidades	1,6	Personalizar mis rutinas	Alcanzar objetivos específicos de entrenamiento	1	5 dias	1,5	1	Terminado
7	Usuario	Seguir las rutinas personalizadas	Alcanzar mis objetivos de manera eficaz	1,7	Seguir mis rutinas personalizadas	Cumplir con mi plan de entrenamiento	2	5 dias	1,6	2	Terminado
8	Usuario	Ajustar preferencias de notificaciones, horarios, etc.	Tener un plan organizado y recibir recordatorios	1,8	Configurar mis preferencias	Recibir notificaciones y recordatorios personalizados	2	6 dias	1,7	2	Terminado
10	Usuario	Acceder a un menú con mis rutinas personalizadas	Tener acceso rápido a mi plan de entrenamiento	1,9	Acceder al menú de rutinas personalizadas	Seleccionar la rutina diaria personalizada que quiero seguir	2	5 dias	1,8	2	Terminado
11	Usuario	Recibir notificaciones sobre mis rutinas y objetivos	Mantenerme motivado y organizado	1,10	Recibir notificaciones de mi plan	Recordar los entrenamientos programados	2	3 dias	1,9	2	Terminado
12	Usuario	Registrar y visualizar calorías gastadas y tiempo de entrenamiento	Monitorizar mi progreso de manera efectiva	1,11	Seguir mi progreso	Ver mis avances en el tiempo	2	4 dias	1,10	2	Terminado
13	Usuario	Crear un calendario con rutinas personalizadas y programarlas	Mantener un plan de entrenamiento organizado y seguirlo semanalmente	1,12	Crear un calendario semanal	Planificar mis rutinas para la semana	1	5 dias	1,11	2	Terminado
14	Administrador	Generar retos semanales	Definir retos segun progreso	2,1	Administrar retos semanales de entrenamiento	Llevar el control de rendiminto de los participantes	1	6 dias	Ninguna	3	Terminado
15	Administrador	Recopilar y mostrar puntajes	Evaluar el puntaje de los retos semanales	2,2	Visualizar panel de retos	Ver puntajes de todos los retos	1	5 dias	2,1	3	Terminado
16	Administrador	Asegurar la visibilidad y relevancia diaria de los mesajes a los participantes	Los mensajes de entrenamiento se destaquen diariamente	2,3	Mostrar mensajes motivacionales	Los mensajes aparezcan de manera destacada diariamente	1	5 dias	2,2	3	Terminado
17	Administrador	Acceder y controlar el sistema donde se controlan todos los datos	Tener acceso para hacer modificaciones de acuerdo a las solicitudes recibidas	2,4	Gestionar datos de todos los participantes	Modicicar y eliminar los datos de los participantes	1	5 dias	2,3	3	Terminado
18	Administrador	Exportar los datos de los usuarios y rutinas	Tener informes detallados y análisis externos	3,1	Generar reportes descargables en formato CSV o PDF	Los reportes deben contener datos filtrables por fecha y categoría	2	5 dias	2,4	3	Terminado
19	Administrador	Gestionar comentarios y feedback de los usuarios	Mejorar la experiencia del usuario y el contenido de la aplicación	3,2	Visualizar y moderar comentarios enviados por los usuarios	Debe existir un panel para moderar el contenido de forma sencilla	2	6 dias	3,1	4	Terminado
20	Administrador	Crear y gestionar múltiples administradores	Delegar tareas administrativas a otros usuarios autorizados	3,3	Definir roles y permisos para diferentes administradores	El sistema debe permitir asignar diferentes niveles de acceso a cada administrador	1	4 dias	3,2	4	Terminado

4.5. Cronograma de Actividades

Figura 11. Cronograma de actividades para Trainup

Nombre de tarea	Nombre	Duración	Comienzo	Fin
Trainup	NOMBRE DEL PROYECTO: Trainup	81 día?	20-Ago-24	21-Nov-24
Sprint 1: Configuración Inicial y Planificación (Semana 1-2)	1.0 INICIO	15 días	21-Ago-24	22-Nov-24
Semana 1:Definición de Requerimientos: Finalizar los requerimientos funcionales y técnicos.	Crear la visión del proyecto	2 día	22-Ago-24	23-Nov-24
Configuración del Entorno de Desarrollo: Configurar Git, Docker, y los repositorios.	Identificar los roles clave y stakeholders	2 día	23-Ago-24	24-Nov-24
Semana 2:Configuración del Proyecto: Crear la estructura inicial del proyecto utilizando el stack MERN.	Crear acta de constitución del proyecto	2 día	24-Ago-24	25-Nov-24
Implementación de Autenticación Básica:Registro de Usuario: Permitir la creación de cuentas con nombre, correo electrónico y contraseña.	Formar equipo Scrum	2 día	25-Ago-24	26-Nov-24
Inicio de Sesión: Autenticar a los usuarios mediante correo electrónico y contraseña.	Desarrollar Epicas	2 día	26-Ago-24	27-Nov-24
Sprint 2: Desarrollo del Backend (Semana 3-4)	Crear el backlog priorizado del producto	2 día	27-Ago-24	28-Nov-24
Semana 3:Diseño de la Base de Datos: Modelar las colecciones de MongoDB (usuarios, rutinas, estadísticas).	Crear la planificación de lanzamiento	1 día	28-Ago-24	29-Nov-24
Desarrollo de APIs Backend:APIs para Registro de Usuario e Inicio de Sesión.	2.0 PLANIFICACION Y ESTIMACION	21 días	29-Ago-24	30-Nov-24
APIs para la selección de nivel de dificultad y generación de rutinas básicas.	Crear historias de usuario	2 días	30-Ago-24	1-Dic-24
Semana 4:Implementación del Backend:Configuración de Express.js y Node.js.	Estimar historias de usuario	3 días	31-Ago-24	2-Dic-24
Desarrollo de lógica para CRUD de rutinas.	Crear tareas	4 días	1-Set-24	3-Dic-24
Integración de la lógica de selección de nivel de dificultad (principiante, intermedio, avanzado).	Estimar tareas	5 días	2-Set-24	4-Dic-24
Sprint 3: Desarrollo del Frontend (Semana 5-6)	Crear plan de calidad	6 días	3-Set-24	5-Dic-24
Semana 5:Diseño de la Interfaz de Usuario: Crear mockups para el flujo de registro, inicio de sesión, y selección de rutinas.	Crear el registro de riesgos	7 días	4-Set-24	6-Dic-24
Implementación del Frontend:Desarrollo de componentes para el registro e inicio de sesión en React.js.	Crear el cronograma	8 días	5-Set-24	7-Dic-24
Implementación de la interfaz para la selección de nivel de dificultad.	3.0 IMPLEMENTACION, REVISION Y RETROSP	15 días	6-Set-24	8-Dic-24

Semana 6:Menú de Rutinas:Desarrollo de un menú para que los usuarios seleccionen y sigan una rutina diaria.	SPRINT 1	2 días	7-Set-24	9-Dic-24
Implementación de la visualización de estadísticas de progreso (calorías gastadas, tiempo total de entrenamiento).	Reunión de planeación del sprint	3 días	8-Set-24	10-Dic-24
Sprint 4: Personalización y Pruebas (Semana 7-8)	Crear el backlog del sprint	4 días	9-Set-24	11-Dic-24
Semana 7:Personalización de Rutinas:Permitir a los usuarios personalizar rutinas según objetivos específicos (bajar de peso, tonificar).	Historia de Usuario 1	5 días	10-Set-24	12-Dic-24
Generar rutinas personalizadas basadas en objetivos y nivel de dificultad.	Tarea 1..n	6 días	11-Set-24	13-Dic-24
Semana 8:Gestión de Preferencias y Programación:Permitir a los usuarios ajustar preferencias de notificaciones y horarios.	Pruebas	7 días	12-Set-24	14-Dic-24
Implementar la posibilidad de programar rutinas y establecer la hora de entrenamiento.	Historia de Usuario 2..n	8 días	13-Set-24	15-Dic-24
Pruebas Unitarias y Funcionales: Realizar pruebas de las APIs y componentes del frontend.	Tarea 1..n	9 días	14-Set-24	16-Dic-24
Sprint 5: Funcionalidades Adicionales, Integración y Despliegue (Semana 9-12)	Pruebas	21 días	15-Set-24	17-Dic-24
Semana 9:Menú de Rutinas Personalizadas:Desarrollar un menú donde los usuarios puedan seleccionar y seguir rutinas personalizadas creadas	Pruebas (testing)	11 días	16-Set-24	18-Dic-24
Notificaciones:Implementar el envío de recordatorios y notificaciones según preferencias del usuario.	Pruebas de Regresión	1 día?	17-Set-24	19-Dic-24
Semana 10:Creación de Calendario Semanal:Permitir a los usuarios crear un calendario semanal para programar rutinas personalizadas para cada día.	Pruebas de Aceptación de Usuario (UAT)	1 día?	18-Set-24	20-Dic-24
Semana 11:Pruebas de Integración Finales: Asegurar la correcta interacción entre todas las funcionalidades.	Pruebas de Desempeño + Seguridad	1 día?	19-Set-24	21-Dic-24
Documentación: Completar la documentación del código y de la aplicación.	Reunión de demostración y revisión del Sprint	1 día?	20-Set-24	22-Dic-24
Semana 12:Despliegue en Producción: Utilizar Docker para desplegar la aplicación en el entorno de producción.	Reunión de retrospectiva del sprint	1 día?	30 junio 2015	23-Dic-24

Capacitación y Soporte: Proveer documentación y soporte para los usuarios finales.	SPRINT 2..n	1 día?	30 junio 2015	24-Dic-24
Entrega Final: Preparar la aplicación para su entrega y lanzamiento.	Reunión de planeación del sprint	1 día?	30 junio 2015	25-Dic-24

4.6. Gestión de Riesgos

Registro de Riesgos.xlsx

Figura 12. Registro de Riesgos

Registro de Riesgos

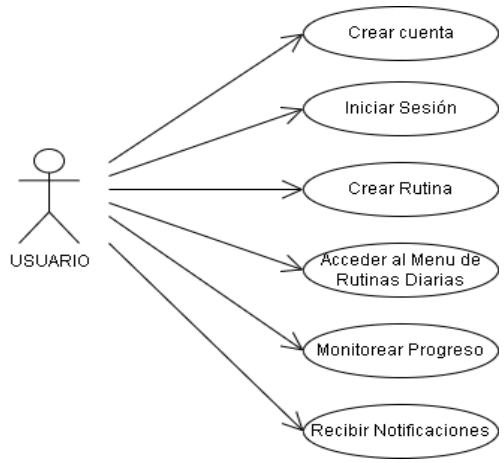
Nombre del Proyecto: Trainup

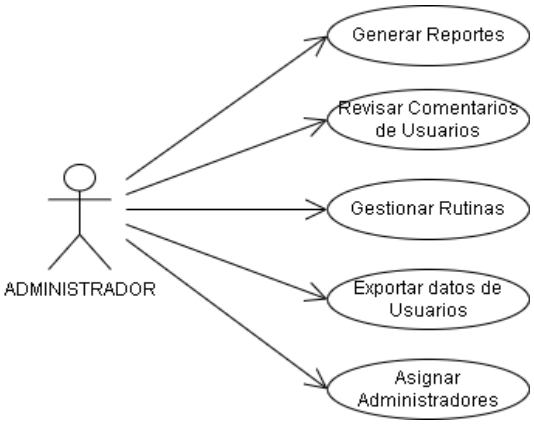
ID Riesgo	Descripción del Riesgo	Area de Impacto	Causa	Impacto	Probabilidad	Puntuación de Riesgo	Detectabilidad	Estado	Asignado a	Evento que lo Dispara	ID Riesgo	Estrategia de Respuesta	Reserva de Contingencia; Ajustes al Presupuesto y Cronograma en el Plan de Gestión del Proyecto	Fecha de Aprobación	Comentarios
1	Nuevas tecnologías representan un reto para el grupo debido a la curva de aprendizaje, lo que podría generar retrasos en las implementaciones.	Cronograma	Falta de experiencia en implementacion de nuevas tecnologias de los integrantes	Muy Serio	Medio	3,5	Bajo	Activo	Equipo de gestión de proyectos y equipo de ventas/cliente	Uso de tecnologías nuevas en el proyecto+R6R16M4:R8M4:RM4:R11	1	Mitigación	Ajuste de cronograma para incluir tiempo de capacitación	14-ago-24	Incluir mentorías técnicas para acelerar el aprendizaje
2	Errores de programación y problemas técnicos que impactan el rendimiento del software.	Calidad	Falta de experiencia en implementacion de nuevas tecnologias de los integrantes	Moderado	Medio	1,5	Medio	Observado	Administradores de sistemas y equipo de soporte técnico	Desarrollo de nuevas funcionalidades sin pruebas exhaustivas	2	Mitigación	Revisión del cronograma y presupuesto para incluir tiempo adicional para pruebas	21-ago-24	Implementar pruebas automáticas revisiones de código
3	Modificaciones constantes en los requisitos del software a lo largo del	Cronograma	Falta de comunicacion efectiva entre usuario y equipo de desarrollo	Serio	Muy Alto	4,5	Alto	Cerrado	Equipo de gestión de proyectos	Revisión de requisitos en fases avanzadas del proyecto	3	Prevención	Ajuste en el cronograma para revisiones frecuentes de los requisitos	28-ago-24	Asegurar reuniones de seguimiento con el cliente
4	Subestimación de los costos involucrados en el desarrollo del aplicativo.	Costo	falta de experiencia en implementacion cotizacion del proyecto	Muy Serio	Bajo	2,1	Bajo	Observado	Todos los miembros del equipo	Actualización del presupuesto inicial tras nuevas estimaciones	4	Mitigación	Ajustar presupuesto basado en evaluaciones trimestrales	4-Set-24	Revisión continua de los costos y márgenes
5	Aumentos inesperados en los costos de desarrollo debido a cambios en los requisitos o tecnologías.	Costo	Falta de comunicacion efectiva entre usuario y equipo de desarrollo	Muy Serio	Alto	4,9	Medio	Observado	Equipo de desarrollo	Solicitudes de cambio por parte del cliente	5	Transferencia	Reservar una contingencia del 10% en el presupuesto	19-Set-24	Establecer límites claros sobre modificaciones aceptables
6	Expansión no controlada de las características y funcionalidades del aplicativo durante el desarrollo.	Alcance	Falta de experiencia en implementacion de nuevas tecnologias de los integrantes	Moderado	Bajo	0,9	Alto	Activo	Equipo de desarrollo	Solicitudes adicionales sin previa planificación	6	Mitigación	Ajuste de alcance sin afectar el presupuesto original	26-Set-24	Implementar control de cambios riguroso
7	Extraviar datos del cliente o de toda la historia del usuario.	Alcance	falla del sistema o servidor – copia de seguridad	Muy Serio	Bajo	2,1	Bajo	Observado	Equipo de desarrollo	Fallo de servidores o problemas técnicos graves	7	Mitigación	Revisión del plan de contingencia para pérdida de datos	2-oct-24	Asegurar replicación de datos diari
8	Incumplimiento de las fechas limites para lograr el objetivo	Cronograma	Falta de coordinacion del equipo de desarrollo del proyecto	Serio	Medio	2,5	Medio	Observado	Gerente del proyecto	Reuniones no efectivas de planificación	8	Mitigación	Ajustar el cronograma para mejorar plazos	9-oct-24	Incorporar herramientas de gestión de tiempos
9	Entrega erronea al no verificar la actualización de documentos	Calidad	Falta de coordinacion del equipo de desarrollo del proyecto	Serio	Bajo	1,5	Medio	Observado	Equipo de gestión de proyectos	Error humano en la gestión de documentos	9	Prevención	Ajustar cronograma para incluir revisiones adicionales	23-oct-24	Crear un checklist obligatorio para las entregas
10	Falta de pruebas puede que el software presente errores y que sea poco confiable	Calidad	Tiempo de entrega del proyecto cerca al tiempo limite	Serio	Medio	2,5	Medio	Observado	Equipo de gestión de proyectos	Plazos cortos que reducen las pruebas	10	Mitigación	Incluir más tiempo en el cronograma para pruebas exhaustivas	30-oct-24	Implementar pruebas continuas en proceso
11	Sobre estimación del presupuesto	Costo	falta de experiencia en implementacion cotizacion del proyecto	Muy Serio	Bajo	2,1	Medio	Activo	Gerente del proyecto	Revisión de costos posteriores	11	Mitigación	Ajustar presupuesto después de cada fase crítica	6-Set-24	Revisión trimestral de estimaciones presupuestarias
12	Dificultad en la adaptabilidad del usuario al usar el aplicativo movil del GYM	Calidad	falta de experiencia en utilizacion del sistema en los usuarios	Muy Serio	Bajo	2,1	Medio	Observado	Gerente del proyecto	Problemas en la adopción del sistema por usuarios	12	Mitigación	Ajuste del presupuesto para capacitación	13-Set-24	Realizar encuestas de satisfacción post-implementación
13	Posibilidad de fallo inesperado del sistema podria resultar una	Calidad	falla del sistema o servidor – copia de seguridad	Critico	Bajo	2,7	Alto	Observado	Equipo de desarrollo	Problemas técnicos graves o falta de backups	13	Mitigación	Asignar presupuesto para soluciones de respaldo	20-Set-24	Implementar estrategias de recuperación ante desastres

CAPÍTULO 5
DISEÑO DEL SISTEMA DE INFORMACIÓN

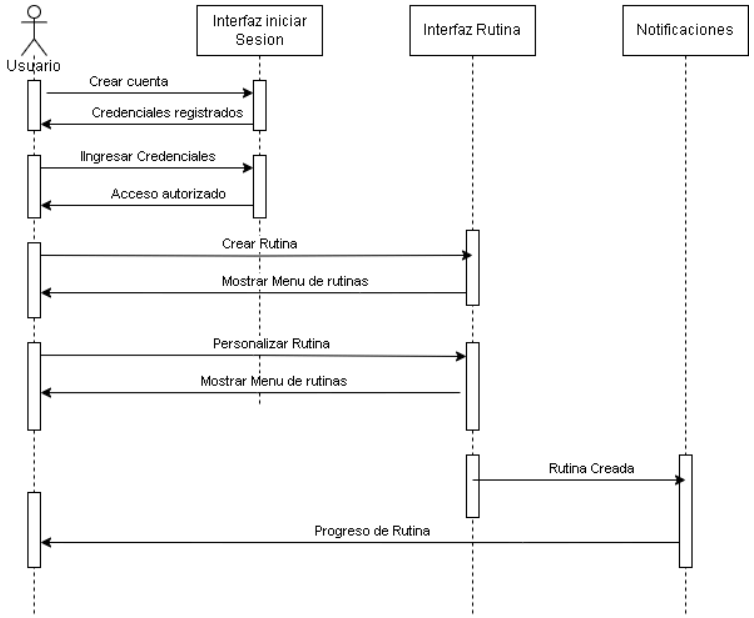
5.1. Diseño de Diagramas UML

5.1.1. Diagramas de casos de uso



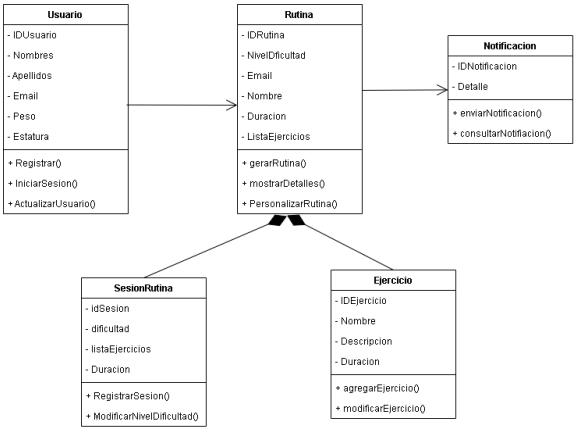


5.1.2. Diagramas de secuencia



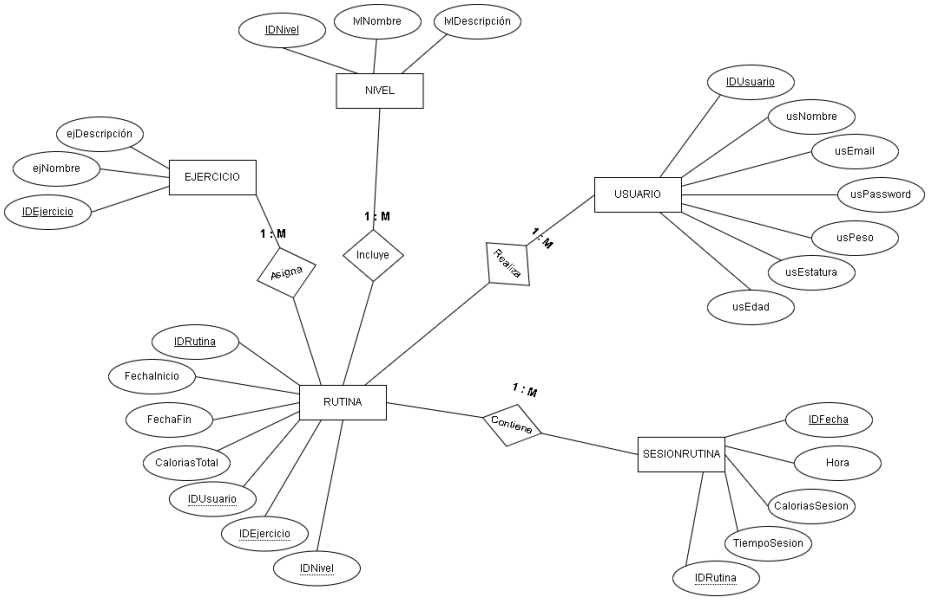
5.1.3. Diagramas de colaboración

5.1.4. Diagramas de clases

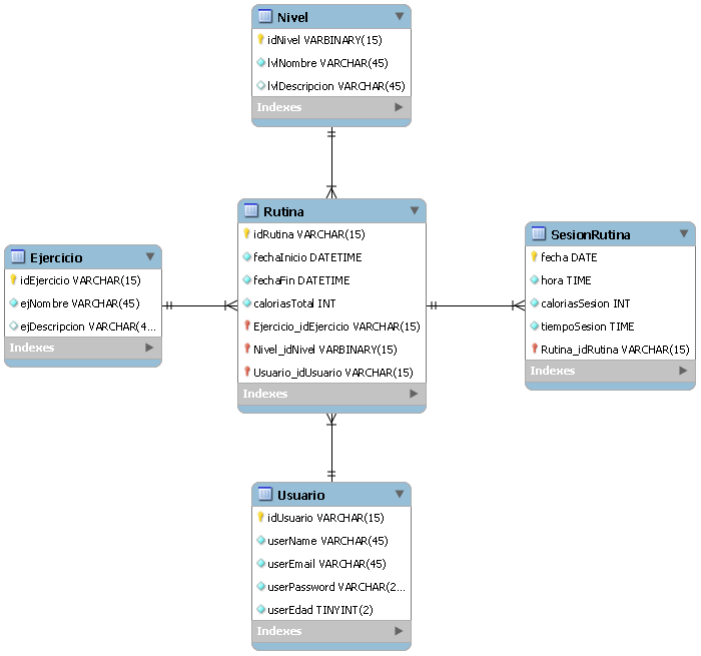


5.2. Diseño de Base de Datos

5.2.1. Diseño conceptual (E/R)



5.2.2. Diseño lógico



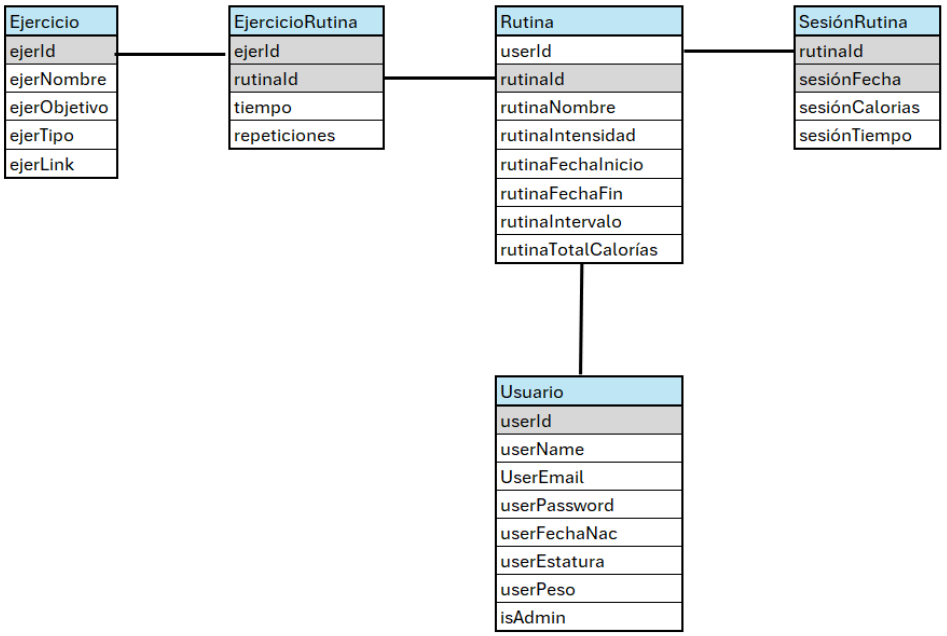
5.2.3. Diseño físico

```sql
CREATE TABLE Ejercicio
(
    idEjercicio VARCHAR(15),
    ejNombre VARCHAR(20),
    ejDescripcion VARCHAR(200),
    CONSTRAINT PK_Ejercicio_idEjercicio PRIMARY KEY (idEjercicio)
)
GO
CREATE TABLE Nivel
(
    idNivel VARCHAR(10),
    lvlNombre VARCHAR(15),
    lvlDescripcion VARCHAR(50),
    CONSTRAINT PK_Nivel_idNivel PRIMARY KEY (idNivel)
)
GO
CREATE TABLE Usuario
(
    idUsuario VARCHAR(12),
    usName VARCHAR(20),
    usEmail VARCHAR(30),
    usPassword VARCHAR(15),
    usPeso NUMERIC(7,3),
    usEdad INT,
    usEstatura INT,
    CONSTRAINT PK_Usuario_idUsuario PRIMARY KEY (idUsuario)
)
GO
CREATE TABLE Rutina
(
    idRutina VARCHAR(15),
    fechaInicio DATETIME,
    FechaFin DATETIME,
    caloriasTotal INT,
    idUsuario VARCHAR(12),
    idNivel VARCHAR(10),
    idEjercicio VARCHAR(15),
    CONSTRAINT PK_Rutina_idRutina PRIMARY KEY (idRutina),
```

Messages

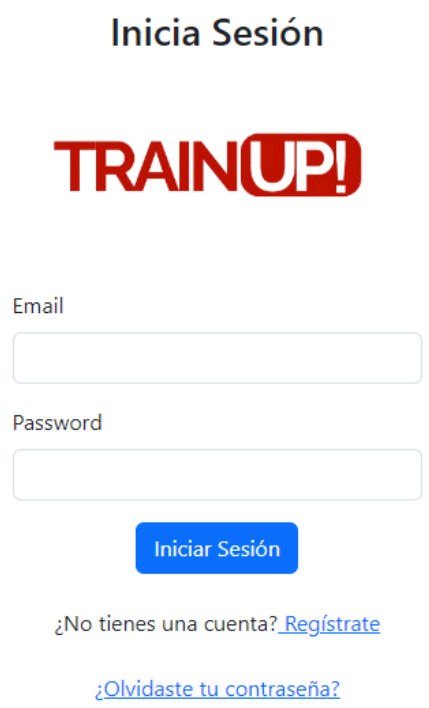
```
Commands completed successfully.

Completion time: 2024-11-24T06:49:25.6939122-05:00
```

5.2.4. Modelado de base de datos

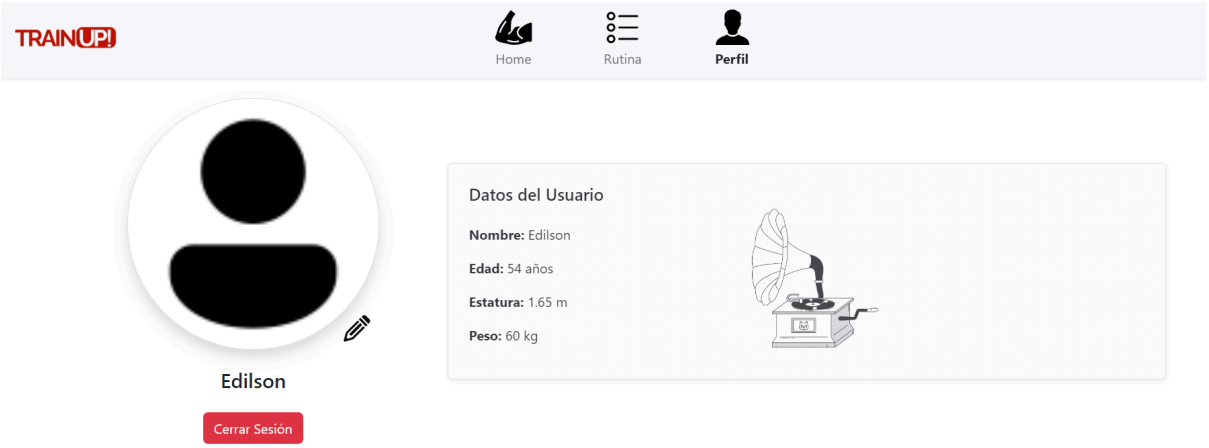


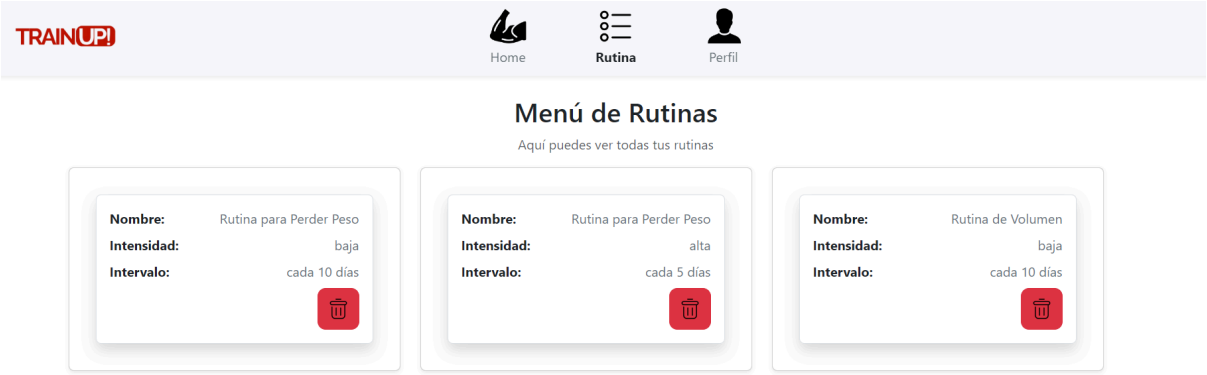
5.3. Diseño de Interfaces Básicas

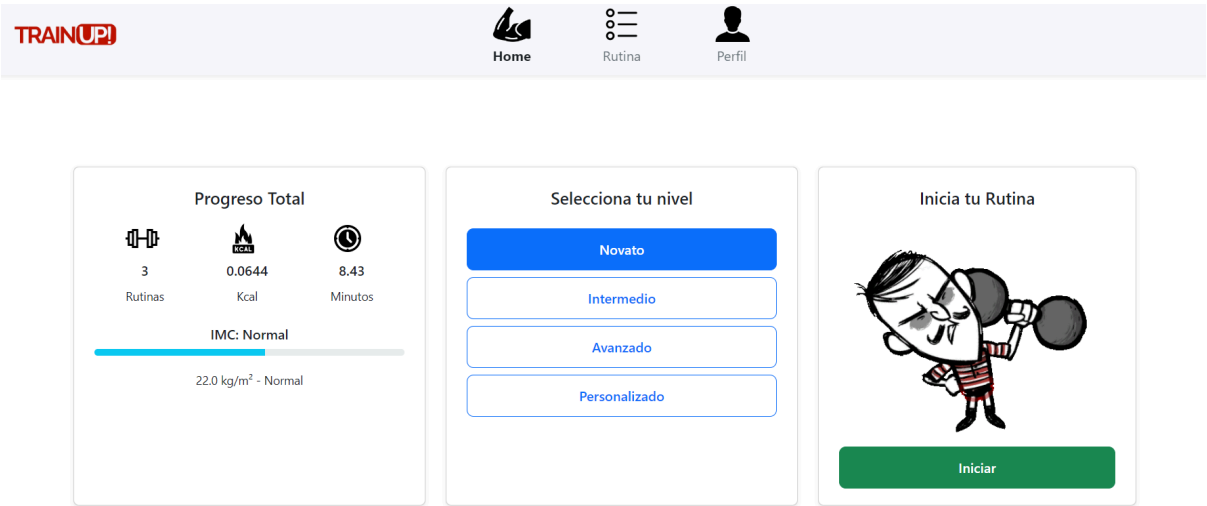
5.3.1. Acceso login



5.3.2. Interfaz …







CAPÍTULO 6
CODIFICACIÓN DEL SOFTWARE

Desarrollo del Sprint 1

6.1.1. Sprint planning

Objetivos:

- Definir y priorizar las tareas necesarias para implementar las funcionalidades básicas del sistema.
- Asegurar que el equipo comprenda claramente los requisitos y criterios de aceptación de cada historia de usuario.

- Establecer un plan de trabajo detallado para los próximos 25 días, incluyendo la asignación de tareas y la estimación de tiempos.

Participantes: Equipo de desarrollo, Scrum Master, Product Owner.

Actividades:

➔ Revisión del backlog del producto.

➔ Selección de historias de usuario prioritarias.

➔ Estimación de tiempos y recursos.

➔ Definición de metas del sprint.

6.1.2. Sprint backlog

	Duración del Sprint	25			**Esfuerzo**
	Tendencia calculada en los últimos	14	**Días**	**Totales**	**140**
ID Tarea	**Tarea**	**ID Historia**	**Responsable**	**Estado**	**Est.**
1.1	Planificación del Proyecto	0	Franz	Terminado	15
1.2	Asignación de Roles y Recursos	0	Angela	Terminado	17
1.3	Criterios de Aceptación	0	Jean	Terminado	12
1.4	Documentación y Roadmap	0	Jose	Terminado	18
1.5	Configuración Inicial del Proyecto	0	Edison	Terminado	15
1.6	Registro de Usuario	HU01	Xiomara	Terminado	17
1.7	Inicio de Sesión	HU02	Isaac	Terminado	12
1.8	Selección de Nivel de Dificultad	HU03	Heison	Terminado	18
1.9	Generación de Rutinas Básicas	HU04	Jean	Termina	16

				do	

6.1.3. Historias de usuarios

ID Historia	Descripción	Objetivo
HU01	Registro de Usuario	Permitir el registro de nuevos usuarios.
HU02	Inicio de Sesión	Habilitar acceso seguro a las cuentas.
HU03	Selección de Nivel de Dificultad	Configurar las preferencias del usuario.
HU04	Generación de Rutinas Básicas	Crear entrenamientos personalizados.

6.1.4. Taskboard

Estado	Tareas	Historia	Responsable(s)
Terminado	Configuración inicial del proyecto.	N/A	Edison, Hieyson
Terminado	Implementar formulario de registro.	HU01	Xiomara, Jose
Terminado	Validación de campos obligatorios en el registro.	HU01	Xiomara, Jose
Terminado	Implementar funcionalidad de conversión de unidades (cm a in, kg a lbs).	HU01	Xiomara, Jose
Terminado	Crear formulario de inicio de sesión.	HU02	Isaac, Jhon Henry
Terminado	Validar credenciales en el inicio de sesión.	HU02	Isaac, Jhon Henry
Terminado	Crear interfaz de selección de nivel de dificultad.	HU03	Heison, Jhon Henry

Terminado	Almacenar el nivel de dificultad seleccionado.	HU03	Heison, Jhon Henry
Terminado	Diseñar lógica para generación de rutinas según el nivel de dificultad.	HU04	Jean Esteven, Franz
Terminado	Crear interfaz para mostrar rutinas generadas.	HU04	Jean Esteven, Franz

6.1.5. Daily scrum

Elemento	Detalle
Participantes	Equipo del proyecto.
Frecuencia	Diario.
Medio	Meet
Duración	15 minutos.
Enfoque	Actualizaciones: ¿Qué hiciste ayer?, ¿Qué harás hoy?, ¿Obstáculos?

6.1.6. Sprint review

¿Qué aprendimos?

- La funcionalidad de registro debe considerar las diferentes unidades de medida utilizadas en otros países, como libras (lbs) en lugar de kilogramos (kg) o pulgadas (in) en lugar de centímetros (cm).
- Es importante planificar sesiones de familiarización técnica para los miembros del equipo que no dominan completamente las herramientas o tecnologías (como React).
- La configuración del entorno puede presentar desafíos si no se estandariza o documenta adecuadamente.

¿Qué estamos haciendo bien?

- Completar todas las tareas del Sprint según el backlog, cumpliendo con los plazos establecidos.
- Colaboración efectiva entre los miembros del equipo para resolver problemas rápidamente.
- Buen manejo de las reuniones diarias (Daily Scrum), permitiendo identificar bloqueos de manera oportuna.

¿Qué podemos hacer mejor?

- Mejorar la planificación del entorno de desarrollo para reducir problemas de configuración inicial.
- Implementar documentación clara para procedimientos clave, como configuraciones específicas del proyecto.
- Asegurarnos de incluir pruebas exhaustivas para escenarios internacionales, como unidades de medida.

Categoría	Observaciones	Acciones a Realizar
Personas	- Algunos miembros no estaban familiarizados con React, lo que ralentizó la configuración inicial.	- Planificar una sesión de capacitación básica sobre React y el entorno del proyecto.
Relaciones	- Colaboración positiva entre los miembros para superar los problemas técnicos.	- Mantener la buena comunicación en el equipo mediante reuniones breves y eficientes.
Procesos	- La falta de estandarización en la configuración del entorno generó retrasos.	- Crear una guía de configuración paso a paso para futuros desarrolladores.
Herramientas	- La funcionalidad de conversión de unidades fue implementada exitosamente, pero debe ser probada más a	- Añadir más pruebas automatizadas para verificar las conversiones entre cm/in y kg/lbs.

	fondo.	

Acciones a Realizar

Funcionalidad de Conversión:

- Asegurarse de que la conversión entre unidades de medida (cm a in, kg a lbs) esté completamente funcional y probada.
- Incluir un selector de unidades en el formulario de registro para que los usuarios elijan su preferencia.

Capacitación Técnica:

- Realizar una sesión de introducción a React para los miembros menos experimentados.
- Documentar las configuraciones de entorno necesarias para estandarizar procesos.

6.1.7. Criterios de aceptación

Historia	Criterios de Aceptación
HU01	- Los usuarios pueden registrarse correctamente.
	- Validaciones en campos obligatorios (correo, contraseña).
	- Al registrarse exitosamente, el sistema debe redirigir al usuario a la pantalla de inicio.
HU02	- Acceso solo permitido con credenciales válidas.
	- Mensajes de error claros si las credenciales son inválidas.
HU03	- Los usuarios deben poder elegir entre niveles (básico, intermedio, avanzado).
	- La selección debe guardarse en su perfil para rutinas futuras.
HU04	- Rutinas básicas generadas según nivel y preferencias iniciales.
	- Cada rutina debe incluir ejercicios válidos y una duración sugerida.

6.1.8. Resultados del sprint

6.1.8.1. Evidencias.

6.1.8.2. **Prueba de desarrollo:** Pruebas unitarias, pruebas de integración, pruebas de aceptación por el usuario.

Desarrollo del Sprint 2

Objetivos: Definir las tareas y objetivos del sprint.

- Planificar la implementación de las funcionalidades relacionadas con la personalización de rutinas y la gestión de preferencias de los usuarios.
- Revisar y ajustar las prioridades del backlog del producto según el feedback recibido durante el Sprint 1.
- Asegurar que el equipo tenga una comprensión clara de las nuevas historias de usuario y sus criterios de aceptación.
- Establecer un plan de trabajo detallado para los próximos 25 días, incluyendo la asignación de tareas y la estimación de tiempos.

Participantes: Equipo de desarrollo, Scrum Master, Product Owner.

Actividades:

- ➔ Revisión del backlog del producto.
- ➔ Selección de historias de usuario prioritarias.
- ➔ Estimación de tiempos y recursos.
- ➔ Definición de metas del sprint.

6.1.9. Sprint backlog

	Duración del Sprint	25		
	Tendencia calculada en los últimos	14	Días	Totales
ID Tarea	Tarea	ID Historia	Responsable	Estado
2	Menú de Rutinas	HU05	Franz	Terminado
2.1	Personalización de Rutinas	HU06	Angela	Terminado
2.2	Gestión de Preferencias y Programación	HU07	Jean	Terminado
2.3	Menú de Rutinas Personalizadas	HU08	Jose	Terminado

6.1.10. Historias de usuarios

ID Historia	Historia de Usuario (HU)
HU05	Menú de Rutinas
HU06	Personalización de Rutinas
HU07	Gestión de Preferencias y Programación
HU08	Menú de Rutinas Personalizadas

6.1.11. Taskboard

Estado	Tarea	Historia
Terminado	Implementar registro de usuario	HU01
Terminado	Implementar inicio de sesión	HU02
Terminado	Implementar selección de nivel de dificultad	HU03
Terminado	Implementar generación de rutinas básicas	HU04
Terminado	Menú de Rutinas	HU05
Terminado	Personalización de Rutinas	HU06
Terminado	Gestión de Preferencias y Programación	HU07
Terminado	Menú de Rutinas Personalizadas	HU08

6.1.12. Daily scrum

Elemento	Detalle
Participantes	Equipo del proyecto.
Frecuencia	Diario.
Medio	Meet
Duración	15 minutos.
Enfoque	Actualizaciones: ¿Qué hiciste ayer?, ¿Qué harás hoy?, ¿Obstáculos?

6.1.13. Sprint review

¿Qué aprendimos?

- La interfaz debe ser adaptable no solo a dispositivos móviles, sino también a PCs. Es esencial que el diseño sea más responsivo.
- La configuración del entorno puede presentar desafíos si no se estandariza o documenta adecuadamente.

¿Qué estamos haciendo bien?

- Completar todas las tareas del Sprint según el backlog, cumpliendo con los plazos establecidos.
- Colaboración efectiva entre los miembros del equipo para resolver problemas rápidamente.
- Buen manejo de las reuniones diarias (Daily Scrum), permitiendo identificar bloqueos de manera oportuna.

¿Qué podemos hacer mejor?

Mejorar la planificación del entorno de desarrollo para reducir problemas de configuración inicial.

Implementar documentación clara para procedimientos clave, como configuraciones específicas del proyecto.

Asegurarnos de incluir pruebas exhaustivas para diferentes escenarios.

Categoría	Observaciones	Acciones a Realizar
Relaciones	Colaboración positiva entre los miembros para superar los problemas técnicos.	Mantener la buena comunicación en el equipo mediante reuniones breves y eficientes.
Procesos	La falta de estandarización en la configuración del entorno generó retrasos.	Crear una guía de configuración paso a paso para futuros desarrolladores.
Herramientas	La funcionalidad de conversión de unidades fue implementada exitosamente, pero debe ser probada más a fondo.	Añadir pruebas de usabilidad y asegurarse de que el diseño sea responsivo para dispositivos móviles y de escritorio.

Mejorar la comunicación interna: Mantener reuniones diarias breves y eficientes para identificar bloqueos y asegurarse de que todos estén alineados.

Optimización de diseño: Asegurarse de que el diseño sea completamente responsivo, adaptándose bien tanto a dispositivos móviles como a PC.

Pruebas de usabilidad: Implementar pruebas de usabilidad para asegurarse de que la interfaz sea intuitiva y fácil de usar para los usuarios finales.

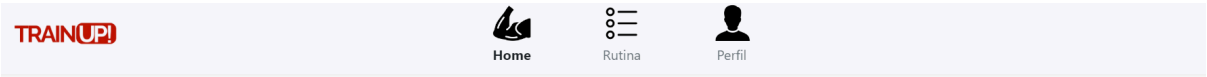
6.1.14. Criterios de aceptación

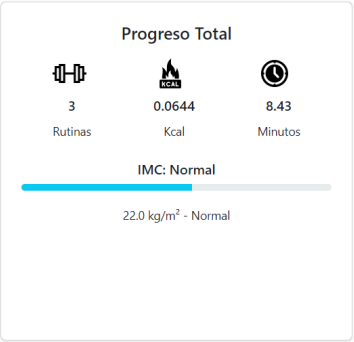
ID Historia	Historia de Usuario (HU)	Criterios de Aceptación

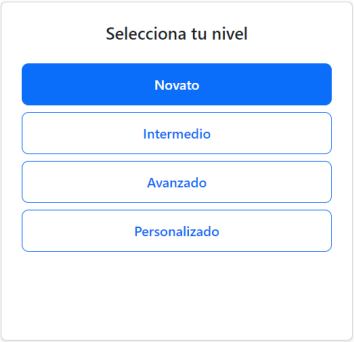
HU01	Implementar selección de nivel de dificultad	- El usuario puede seleccionar entre al menos tres niveles de dificultad. - La selección se guarda correctamente en el perfil del usuario.
HU02	Implementar generación de rutinas básicas	- El sistema genera una rutina básica automáticamente basada en el nivel de dificultad seleccionado. - La rutina generada incluye ejercicios variados y adecuados para el nivel.
HU03	Menú de Rutinas	- El menú permite al usuario ver todas las rutinas generadas y personalizadas. - La interfaz es intuitiva y muestra información clara sobre cada rutina.
HU04	Personalización de Rutinas	- El usuario puede personalizar las rutinas generadas, agregando o eliminando ejercicios. - Los cambios en la rutina se guardan correctamente.
HU05	Gestión de Preferencias y Programación	- El usuario puede ajustar sus preferencias, como la duración de las rutinas o el tipo de ejercicios. - Las preferencias se aplican correctamente a las rutinas generadas.
HU06	Menú de Rutinas Personalizadas	- El usuario puede acceder al menú de rutinas personalizadas y ver todas las modificaciones realizadas. - Las rutinas personalizadas se muestran de forma clara y editable.

6.1.15. Resultados del sprint

6.1.15.1. Evidencias.







6.1.15.2. **Prueba de desarrollo:** Pruebas unitarias, pruebas de integración, pruebas de aceptación por el usuario.

6.1.16. Sprint planning

Objetivos:

- Planificar la implementación de las funcionalidades de notificaciones y seguimiento de progreso.
- Revisar y ajustar las prioridades del backlog del producto según el feedback recibido durante el Sprint 2.
- Asegurar que el equipo comprenda claramente los requisitos y criterios de aceptación de las nuevas historias de usuario.
- Establecer un plan de trabajo detallado para los próximos 25 días, incluyendo la asignación de tareas y la estimación de tiempos.

Participantes: Equipo de desarrollo, Scrum Master, Product Owner.

Actividades:

- ➔ Revisión del backlog del producto.
- ➔ Selección de historias de usuario prioritarias.
- ➔ Estimación de tiempos y recursos.
- ➔ Definición de metas del sprint.

ID Tarea	Tarea	ID Historia	Responsable	Estado
3	Notificaciones	HU09	Franz	Terminado
3.1	Seguimiento de Progreso	HU10	Angela	Terminado
3.2	Creación de Calendario Semanal	HU11	Jean	Terminado
3.3	Implementar API para obtener rutinas según el nivel de dificultad	HU08	Jose	Terminado
3.4	Crear una pantalla para que el usuario seleccione su nivel de dificultad	HU08	Edison	Terminado
3.5	Conectar el frontend con la API para la selección de nivel de dificultad	HU08	Xiomara	Terminado
3.6	Implementar la visualización de rutinas básicas según el nivel de	HU09	Edison	Terminado

	dificultad			
3.7	Mejora de Perfil y Autenticación	HU09	Jose	Terminado
3.8	Creación de perfil de administrador	HU19	Eduardo	Terminado

6.1.17. Sprint backlog

	Duración del Sprint	25		
	Tendencia calculada en los últimos	14	**Días**	**Totales**
ID Tarea	**Tarea**	**ID Historia**	**Responsable**	**Estado**
3	Notificaciones	HU09	Franz	Terminado
3.1	Seguimiento de Progreso	HU10	Angela	Terminado
3.2	Creación de Calendario Semanal	HU11	Jean	Terminado
3.3	Implementar API para obtener rutinas según el nivel de dificultad seleccionado.	HU08	Jose	Terminado
3.4	Crear una pantalla para que el usuario seleccione su nivel de dificultad.	HU08	Edison	Terminado
3.5	Conectar el frontend con la API para la selección de nivel de dificultad.	HU08	Xiomara	Terminado
3.6	Implementar la visualización de rutinas básicas según el nivel de dificultad seleccionado.	HU09	Edison	Terminado
3.7	Mejora de Perfil y Autenticación	HU09	Jose	Terminado
3.8	Creacion de perfil de administrador	HU19	Eduardo	Terminado

6.1.18. Historias de usuarios

ID	Historia de Usuario
HU09	Notificaciones

HU10	Seguimiento de Progreso
HU11	Implementar la previsualización de rutinas básicas según el nivel de dificultad seleccionado
HU12	Recuperación de Contraseña
HU13	Verificación en dos pasos

6.1.19. Taskboard

ID	Tareas	Responsable	Estado
HU01	Notificaciones	Edison	Terminado
HU02	Seguimiento de Progreso	Edison	Terminado
HU03	Implementar API para obtener rutinas según el nivel de dificultad seleccionado	Jean	Terminado
HU04	Crear una pantalla para que el usuario seleccione su nivel de dificultad	Jose	Terminado
HU05	Conectar el frontend con la API para la selección de nivel de dificultad	Edison	Terminado
HU06	Implementar la visualización de rutinas básicas según el nivel de dificultad seleccionado	Xiomara	Terminado
HU07	Mejora de Perfil y Autenticación	Edison	Terminado

6.1.20. Daily scrum

Elemento	Detalle
Participantes	Equipo del proyecto.
Frecuencia	Diario.
Medio	Meet
Duración	15 minutos.
Enfoque	Actualizaciones: ¿Qué hiciste ayer?, ¿Qué harás hoy?, ¿Obstáculos?

6.1.21. Sprint review

1. Reflexiones sobre el Proyecto

¿Qué aprendimos?

- Comprendimos la importancia de la seguridad y la usabilidad en el registro de usuario.
- En diseño, aprendimos a aprovechar la retroalimentación para optimizar la experiencia de usuario en la aplicación de entrenamiento.

¿Qué estamos haciendo bien?

- Implementamos validaciones de registro y autenticación en dos pasos, facilitando una experiencia segura.
- La estructura de rutinas por nivel y la personalización ayudan a los usuarios a progresar de manera segura y efectiva.

¿Qué podemos hacer mejor?

Capacitar al equipo en seguridad y UX para mejorar las rutinas y fragmentos personalizados.

Integrar retroalimentación entre equipos, realizar pruebas UI/UX continuas, y optimizar interfaces para una navegación intuitiva.

Categoría	Observaciones	Acciones a Realizar
Relaciones	Colaboración positiva entre los miembros para superar los	Mantener la buena comunicación en el equipo mediante reuniones breves y eficientes.

	problemas técnicos.	
Procesos	La falta de estandarización en la configuración del entorno generó retrasos.	Crear una guía de configuración paso a paso para futuros desarrolladores.
Herramientas	La funcionalidad de conversión de unidades fue implementada exitosamente, pero debe ser probada más a fondo.	Añadir pruebas de usabilidad y asegurarse de que el diseño sea responsivo para dispositivos móviles y de escritorio.
Seguridad	La autenticación en dos pasos ha sido implementada, pero se necesita una validación continua.	Añadir la opción de recuperación de contraseña y realizar pruebas periódicas de seguridad.

Acciones a Realizar

Seguridad: Añadir autenticación en dos pasos y la opción de recuperación de contraseña.

UX: Mejorar las interfaces basadas en las pruebas de usabilidad para una navegación más intuitiva.

Integración de Retroalimentación: Realizar sesiones de retroalimentación entre equipos de diseño y desarrollo para afinar la experiencia del usuario.

Pruebas Continuas: Aumentar las pruebas de seguridad y de experiencia de usuario para garantizar que todas las funcionalidades estén optimizadas y sean seguras.

6.1.22. Criterios de aceptación

ID	Historia de Usuario	Criterios de Aceptación
HU01	Notificaciones	1. El usuario debe recibir notificaciones en tiempo real cuando haya actualizaciones importantes.
		2. Las notificaciones deben ser personalizables, permitiendo al usuario activar o desactivar ciertos tipos de notificaciones.
HU02	Seguimiento de Progreso	1. El progreso del usuario debe ser visible en un gráfico interactivo que muestre su avance en cada rutina realizada.
		2. El usuario debe poder ver un resumen semanal de sus actividades y logros.
HU03	Implementar API para obtener rutinas según el nivel de dificultad seleccionado	1. La API debe retornar rutinas correctamente basadas en el nivel de dificultad especificado por el usuario.
		2. La API debe ser capaz de manejar diferentes niveles de dificultad y devolver datos precisos para cada uno.
HU04	Crear una pantalla para que el usuario seleccione su nivel de dificultad	1. La pantalla debe ofrecer opciones claras de nivel de dificultad (bajo, medio, alto).
		2. El usuario debe poder cambiar el nivel de dificultad en cualquier momento antes de comenzar una rutina.
HU05	Conectar el frontend con la API para la selección de nivel de dificultad	1. El frontend debe enviar correctamente la selección de nivel de dificultad a la API para que se devuelvan las rutinas correspondientes.
		2. El frontend debe actualizar la interfaz según el nivel de dificultad seleccionado por el usuario.
HU06	Implementar la visualización de rutinas básicas según el nivel de dificultad seleccionado	1. Las rutinas deben ser visualizadas de manera clara y estructurada según el nivel de dificultad elegido.
		2. Las rutinas deben ser fáciles de entender, con instrucciones concisas y adecuadas para el nivel seleccionado.
HU07	Mejora de Perfil y Autenticación	1. El perfil del usuario debe permitir la edición de datos básicos, como nombre, correo y foto de perfil.
		2. El sistema debe permitir autenticación segura a través de contraseñas o autenticación de dos factores.

6.1.23. Resultados del sprint

6.1.23.1. Evidencias.

6.1.23.2. **Prueba de desarrollo:**Pruebas unitarias, pruebas de integración, pruebas de aceptación por el usuario.

Desarrollo del Sprint 4

Objetivos:

- Planificar la implementación de las funcionalidades administrativas y cualquier tarea final necesaria para completar el proyecto.
- Revisar y ajustar las prioridades del backlog del producto según el feedback recibido durante el Sprint 3.
- Asegurar que el equipo tenga una comprensión clara de las historias de usuario restantes y sus criterios de aceptación.

- Establecer un plan de trabajo detallado para los próximos 26 días, incluyendo la asignación de tareas y la estimación de tiempos.
- Preparar el equipo para la fase de revisión final y la entrega del proyecto.

Participantes: Equipo de desarrollo, Scrum Master, Product Owner.

Actividades:

- ➔ Revisión del Backlog del Producto.
- ➔ Selección de Historias de Usuario.
- ➔ Estimación de Tiempos y Recursos.
- ➔ Definición de Metas del Sprint.
- ➔ Planificación de Tareas.
- ➔ Identificación de Riesgos y Dependencias.
- ➔ Revisión de Criterios de Aceptación.
- ➔ Configuración del Taskboard.

6.1.24. Sprint planning

ID Tarea	Tarea	Duración de la Tarea	Responsable
HU11	Administrar Retos Semanales	2 días	Franz
HU12	Visualizar Panel de Retos	3 días	Angela
HU16	Mostrar Mensajes Motivacionales	2 días	Jean
HU18	Gestionar Datos de los Participantes	3 días	Jose
HU20	Visualizar y Moderar Comentarios	2 días	Xiomara
HU20	Definir Roles y Permisos	1 día	Heison

6.1.25. Sprint backlog

	Duración del Sprint	25		
	Tendencia calculada en los últimos	14	Días	Totales
ID Tarea	Tarea	ID Historia	Responsable	Estado
	Administrar Retos Semanales	HU11	Franz	Terminado
	Visualizar Panel de Retos	HU12	Angela	Terminado
	Mostrar Mensajes Motivacionales	HU16	Jean	Terminado
	Gestionar Datos de los Participantes	HU18	Jose	Terminado
	Visualizar y Moderar Comentarios	HU20	Xiomara	Terminado
	Definir Roles y Permisos	HU20	Heison	Terminado

6.1.26. Historias de usuarios

ID Historia	Historia de Usuario
HU11	Administrar los retos semanales de manera eficiente y permitir a los usuarios seguir su progreso.
HU12	Visualizar el panel de retos donde los usuarios puedan acceder a la información de sus desafíos.
HU16	Mostrar mensajes motivacionales para apoyar a los usuarios a seguir adelante con sus rutinas.
HU18	Gestionar los datos de los participantes, asegurando que toda la información esté organizada y accesible.
HU20	Definir los roles y permisos adecuados para gestionar la seguridad y las funciones dentro de la aplicación.

6.1.27. Taskboard

ID Tarea	Tarea	ID Historia	Duración de la Tarea	Responsable	Estado
T01	Administrar Retos Semanales	HU11	2 días	Franz	Terminado
T02	Configurar Notificaciones de Retos	HU11	2 días	Franz	Pendiente
T03	Visualizar Panel de Retos	HU12	3 días	Angela	Terminado
T04	Actualizar la Interfaz de Usuarios	HU12	2 días	Angela	Pendiente
T05	Mostrar Mensajes Motivacionales	HU16	2 días	Jean	Terminado
T06	Añadir Feedback Post-Reto	HU16	1 día	Jean	Pendiente
T07	Gestionar Datos de los Participantes	HU18	3 días	Jose	Terminado
T08	Implementar Filtros de Búsqueda	HU18	2 días	Jose	Pendiente
T09	Visualizar y Moderar Comentarios	HU20	2 días	Xiomara	Terminado
T10	Revisar Configuración de Roles	HU20	1 día	Heison	Pendiente
T11	Definir Roles y Permisos	HU20	1 día	Heison	Terminado

6.1.28. Daily scrum

Elemento	Detalle
Participantes	Equipo del proyecto.
Frecuencia	Diario.
Medio	Meet
Duración	15 minutos.
Enfoque	Actualizaciones: ¿Qué hiciste ayer?, ¿Qué harás hoy?, ¿Obstáculos?

6.1.29. Sprint review

Reflexiones sobre el Proyecto

¿Qué aprendimos?

Comprendimos la importancia de la **seguridad** y la **usabilidad** en el registro de usuario.

En diseño, aprendimos a aprovechar la **retroalimentación** para optimizar la experiencia de usuario en la aplicación de entrenamiento.

¿Qué estamos haciendo bien?

Implementamos **validaciones de registro** y **autenticación en dos pasos**, facilitando una experiencia segura.

La **estructura de rutinas por nivel** y la **personalización** ayudan a los usuarios a progresar de manera segura y efectiva.

¿Qué podemos hacer mejor?

Capacitar al equipo en **seguridad** y **UX** para mejorar las rutinas y fragmentos personalizados.

Integrar **retroalimentación entre equipos**, realizar **pruebas UI/UX continuas**, y optimizar interfaces para una navegación más intuitiva.

Acciones a Realizar

Integración de Retroalimentación: Realizar sesiones de retroalimentación entre equipos de **diseño** y **desarrollo** para afinar la experiencia del usuario.

Pruebas Continuas: Aumentar las **pruebas de seguridad** y de **experiencia de usuario** para garantizar que todas las funcionalidades estén optimizadas y sean seguras.

6.1.30. Criterios de aceptación

ID Historia	Historia de Usuario	Criterios de Aceptación

HU11	Administrar Retos Semanales	1. Los retos deben ser administrables por el equipo, permitiendo añadir, editar y eliminar retos.
		2. Los usuarios deben poder ver los retos activos y su progreso en tiempo real.
HU12	Visualizar Panel de Retos	1. El panel debe ser visualizable en dispositivos móviles y de escritorio.
		2. Los usuarios deben ver información clara sobre los retos actuales y sus logros.
HU16	Mostrar Mensajes Motivacionales	1. Los mensajes deben ser visibles en la pantalla de inicio, con variación según el progreso del usuario.
		2. Los mensajes deben actualizarse al completar un reto o actividad.
HU18	Gestionar Datos de los Participantes	1. Los datos de los participantes deben ser almacenados y actualizados correctamente en la base de datos.
		2. Los administradores deben tener acceso para visualizar y modificar los datos de los participantes.
HU20	Visualizar y Moderar Comentarios	1. Los comentarios deben ser moderados para garantizar que cumplan con las políticas de la comunidad.
		2. Los usuarios deben poder reportar comentarios inapropiados.
HU20	Definir Roles y Permisos	1. Los roles deben ser claramente definidos, con permisos específicos para administradores y usuarios normales.
		2. Los usuarios deben tener acceso solo a las funciones y datos permitidos según su rol.

6.1.31. Resultados del sprint

6.1.31.1. Evidencias.

6.1.31.2. Prueba de desarrollo.

CAPÍTULO 7
PRUEBAS DE SOFTWARE

7.1. Plan de Pruebas

7.1.1. ***Prueba de caja negra de la aplicación***

1. La aplicación debe ser compatible con los diferentes navegadores web como chrome, edge, etc. para garantizar el uso de este mismo.

Caso de prueba de caja negra: La aplicación debe ser compatible con los diferentes navegadores web como chrome, edge, etc. para garantizar el uso de este mismo.	
Descripción:	Verificación de que la aplicación sea compatible con los navegadores más usados como chrome, edge, entre otros.
Precondiciones:	● La aplicación debe estar lista para correr en los distintos navegadores web.
Pasos:	● Iniciar la aplicación web en los distintos navegadores web. ● Verificar que la aplicación funcione de manera fluida. ● Registrar usuarios en la aplicación web ● Iniciar sesión con los usuarios registrados. ● Iniciar rutinas de entrenamientos. ● Editar perfil de usuario. ● Cierre de sesión en la aplicación.
Resultado esperado:	● La aplicación funciona correctamente en los distintos navegadores web.
Criterio de validación:	● Se valida que la aplicación funcione de manera fluida en los distintos navegadores web ● Se valida que las funciones de la aplicación funcionen correctamente y no haya errores en los procesos.
Resultados:	● El caso de prueba se considera

	exitoso, ya que la aplicación funciona de manera normal y fluida en los diferentes navegadores web, por la parte de procesos como registro, inicio de sesión, entre otros funcionan como se esperaba.

2. La IU debe ser intuitiva y fácil de usar para los usuarios finales.

Caso de prueba de caja negra: La IU debe ser intuitiva y fácil de usar para los usuarios finales	
Descripción:	Verificación de que la interfaz de la aplicación sea amigable, intuitiva y fácil de usar para los usuarios
Precondiciones:	● La aplicación debe estar ejecutándose para su prueba o uso.
Pasos:	● Iniciar la aplicación web en un navegador web. ● iniciar sesión. ● Interactuar con el software ya sea editando datos, creando rutinas de entrenamiento, comenzando rutinas de entrenamiento etc. ● Repetir el 3er paso con diferentes usuarios.
Resultado esperado:	● La UI le parece amigable, fácil de usar e intuitiva a los diferentes usuarios
Criterio de validación:	● Se valida que la interfaz de la aplicación sea amigable, fácil de usar e intuitiva a los diferentes usuarios.
Resultados:	● El caso de prueba se considera exitoso, ya que la interfaz de la aplicación les parece amigable, fácil de usar e intuitiva a los diferentes usuarios.

3. Asegurar que los datos de los usuarios estén protegidos mediante encriptación y autenticación segura.

Caso de prueba de caja negra: Asegurar que los datos de los usuarios estén protegidos mediante encriptación y autenticación segura.	
Descripción:	Asegurar que los datos de los usuarios estén protegidos mediante encriptación y autenticación segura.
Precondiciones:	● La aplicación debe estar ejecutándose para su prueba o uso.
Pasos:	● Iniciar la aplicación web en un navegador web. ● iniciar sesión. ● Verificar la existencia de los tokens de la sesión del usuario en el navegador web ● Verificar si las contraseñas en la base de datos se registran como hasheadas. ● Cerrar sesión por usuario. ● Verificar que los tokens de inicio de sesión se hayan eliminado
Resultado esperado:	● Que el navegador contenga los tokens creados por inicio de sesión de los usuarios, y estos después de cerrar sesión también se eliminen. ● Las contraseñas de los usuarios registrados en la base de datos se registran como hasheadas.
Criterio de validación:	● Se validan los tokens por inicio de sesión. ● Se valida que las contraseña se almacenen como hasheadas en la base de datos.
Resultados:	● El caso de prueba se considera exitoso, ya que con cada inicio de sesión se crean los tokens y se eliminan éstos al cerrar sesion en la aplicacion, tambien en la base de datos las contraseñas se almacenan como hasheadas

4. **Garantizar tiempos de respuesta rápidos y una experiencia de usuario fluida.**

Caso de prueba de caja negra: Garantizar tiempos de respuesta rápidos y una experiencia de usuario fluida.	
Descripción:	Asegurar que los tiempos de respuesta de la aplicación sea rápida y no mayor a 3 segundos
Precondiciones:	● La aplicación debe estar ejecutándose para su prueba o uso.
Pasos:	● Iniciar la aplicación web en un navegador web. ● iniciar sesión. ● Realizar varias funciones de la aplicación
Resultado esperado:	● Que la aplicación responda rápido en caso de consultas, creado de rutinas, edición de datos o de navegación en la aplicación
Criterio de validación:	● Se valida que la respuesta de la aplicación con las distintas funciones no sean mayor a 3 segundos
Resultados:	● El caso de prueba se considera exitoso, ya que la aplicación responde de manera rapido a las consultas y demás acciones que se realizó.

5. **PRUEBAS DE CAJA NEGRA POR SPRINT**

- **SPRINT 1**

Caso de prueba de caja negra: Garantizar que las historias de usuario definidas para el sprint 1 funcionen correctamente	
Descripción:	Asegurar que la aplicación registre usuarios, inicie sesión, seleccione el

	nivel de dificultad de la rutina a crear, y estas mismas se generen.
Precondiciones:	● La aplicación debe estar ejecutándose para su prueba o uso.
Pasos:	● Iniciar la aplicación web en un navegador web. ● Registro de usuario ● Inicio de sesión ● Seleccionar nivel de rutina ● Generar rutina
Resultado esperado:	● Que la aplicación responda de manera exitosa registrando usuarios, también que los usuarios registrados puedan iniciar sesión, que los usuarios seleccionen un nivel de rutina y puedan generar la rutina.
Criterio de validación:	● Se valida que la respuesta de la aplicación con las distintas funciones sean exitosas y rapidas
Resultados:	● El caso de prueba se considera exitoso, ya que la aplicación responde de manera efectiva con las funcionalidades de las historias de usuario que se realizó.

- **SPRINT 2**

Caso de prueba de caja negra: Garantizar que las historias de usuario definidas para el sprint 2 funcionen correctamente	
Descripción:	Asegurar que la aplicación muestre el menu de rutinas, permita la personalización de rutinas, gestionar preferencias y programarlas para que

	muestre un menú de rutinas personalizadas por el usuario
Precondiciones:	● La aplicación debe estar ejecutándose para su prueba o uso.
Pasos:	● Visualizar Menú de rutinas ● Personalización de rutinas ● Gestión de Preferencias y programación ● Menú de Rutinas Personalizadas
Resultado esperado:	● Que la aplicación responda de manera exitosa mostrando un menú de rutinas, también que el usuario pueda personalizar las rutinas según su preferencia y programación, así también como la visualización del menú de rutinas personalizadas por el usuario.
Criterio de validación:	● Se valida que la respuesta de la aplicación con las distintas funciones sean exitosas y rapidas
Resultados:	● El caso de prueba se considera exitoso, ya que la aplicación responde de manera efectiva con las funcionalidades de las historias de usuario que se realizó en este sprint.

- **SPRINT 3**

Caso de prueba de caja negra: Garantizar que las historias de usuario definidas para el sprint 3 funcionen correctamente	
Descripción:	Asegurar que la aplicación muestre notificaciones, seguimiento de progreso, también previsualizar las rutinas básicas según el nivel de dificultad seleccionado, también el cambio de contraseña y la verificación en dos pasos.
Precondiciones:	● La aplicación debe estar ejecutándose para su prueba o

	uso.
Pasos:	● Visualizar Notificaciones ● Seguimiento de Progreso ● Previsualización de rutinas básicas según el nivel de dificultad seleccionado.
Resultado esperado:	● Que la aplicación responda de manera exitosa visualizando notificaciones, seguimiento de progreso, rutinas básicas según el nivel de dificultad seleccionado, permite el cambio de contraseña y la verificación en dos paso.
Criterio de validación:	● Se valida que la respuesta de la aplicación con las distintas funciones sean exitosas y rapidas
Resultados:	● El caso de prueba se considera exitoso, ya que la aplicación responde de manera efectiva con las funcionalidades de las historias de usuario definidas para este sprint que se realizó.

- **SPRINT 4**

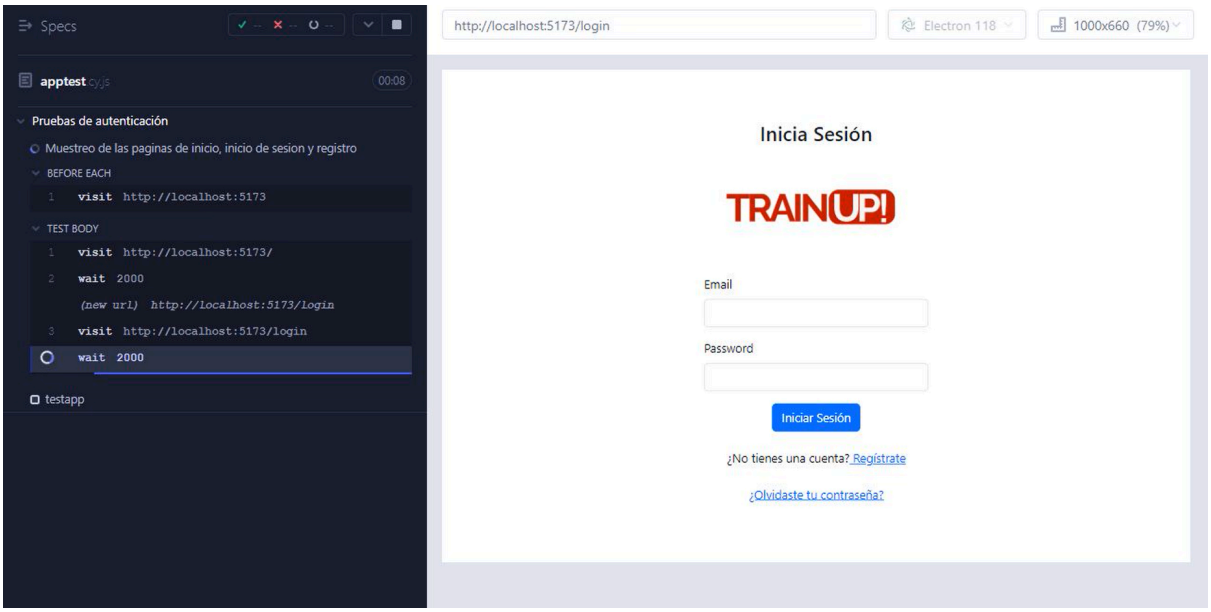
Caso de prueba de caja negra: Garantizar que las historias de usuario definidas para este sprint 4 funcionen correctamente	
Descripción:	Asegurar que la aplicación administre retos semanales de manera eficiente y permita a los usuarios seguir su progreso, visualizar el panel de retos donde los usuarios puedan acceder a la información de sus desafíos, mostrar mensajes motivacionales, gestionar los datos de los participantes, también definir los roles y permisos adecuados para la seguridad de información
Precondiciones:	● La aplicación debe estar ejecutándose para su prueba o uso.
Pasos:	● Administrar los retos semanales de

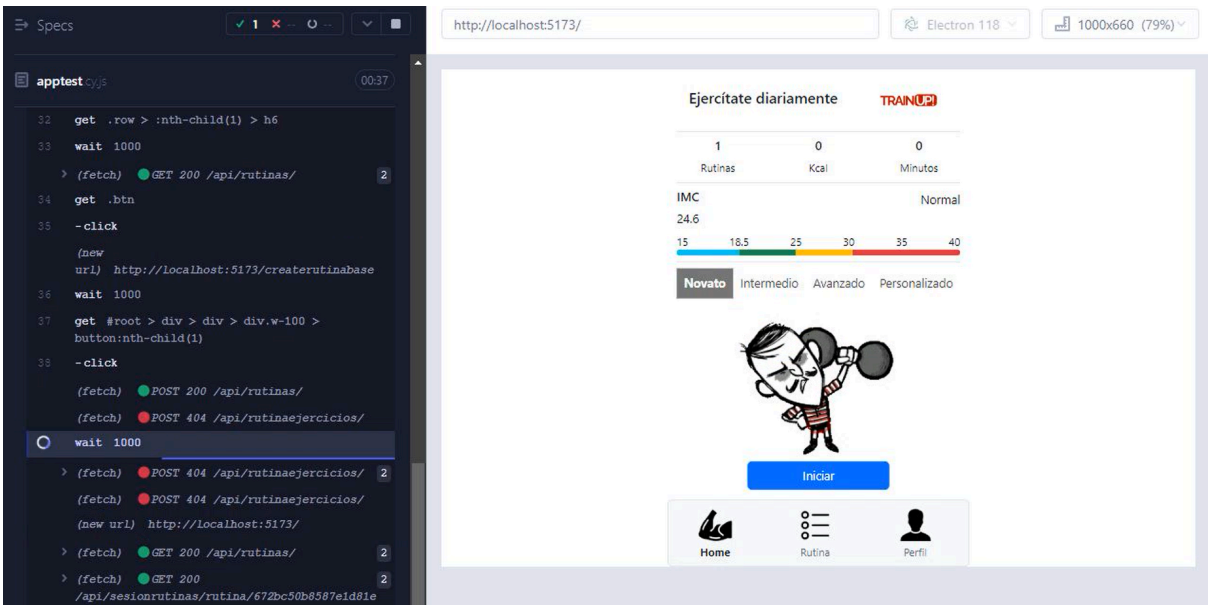
	manera eficiente y permitir a los usuarios seguir su progreso. ● Visualizar el panel de retos donde los usuarios puedan acceder a la información de sus desafíos. ● Mostrar mensajes motivacionales para apoyar a los usuarios a seguir adelante con sus rutinas. ● Gestionar los datos de los participantes, asegurando que toda la información esté organizada y accesible. ● Definir los roles y permisos adecuados para gestionar la seguridad y las funciones dentro de la aplicación.
Resultado esperado:	● Que la aplicación responda de manera exitosa administrar los retos semanales, visualizar el panel de retos, visualizar los mensajes motivacionales, gestión de datos de los participantes y asignar roles y permisos para la seguridad de la información de los usuarios.
Criterio de validación:	● Se valida que la respuesta de la aplicación con las distintas funciones sean exitosas y rapidas
Resultados:	● El caso de prueba se considera exitoso, ya que la aplicación responde de manera efectiva con las funcionalidades de las historias de usuario que se realizó.

7.1.2. **Pruebas de extremo a extremo**

1. **Pruebas E2E Cypress:** Cypress es una herramienta poderosa para realizar pruebas automáticas de aplicaciones web de extremo a extremo.

```
1   describe('Pruebas de autenticación', () => {
2       beforeEach(() => {
3         // visita a la pagina de inicio
4         cy.visit('http://localhost:5173');
5       });
6
7       it('Muestreo de las paginas de inicio, inicio de sesion y registro', () => {
8         cy.visit('http://localhost:5173/');
9         cy.wait(2000)
10        cy.visit('http://localhost:5173/login');
11        cy.wait(2000)
12        cy.visit('http://localhost:5173/register');
13        cy.wait(2000)
14      });
15
16      it('testapp', () => {
17
18        cy.visit('http://localhost:5173/register'); // visita a la pagina de registro
19        cy.wait(2000);
20
21        cy.get('#root > div > form > div:nth-child(3) > input').type('usertest'); // nombre de usuario
22        cy.wait(2000);
23        cy.get('#root > div > form > div:nth-child(4) > input').type('usertest50@mail.com'); // email
24        cy.wait(2000);
25        cy.get('#root > div > form > div:nth-child(5) > input').type('@trainuP999'); // password
26        cy.wait(2000);
27        cy.get('#root > div > form > div:nth-child(6) > input').type('@trainuP999'); // confirmation password
28        cy.wait(2000);
29        cy.get('#root > div > form > div:nth-child(7) > input').type('20'); // edad
30        cy.wait(2000);
31        cy.get('#root > div > form > div:nth-child(8) > div > input').type('171'); // talla
32        cy.wait(2000);
33        cy.get('#root > div > form > div:nth-child(9) > div > input').type('72'); // peso
34        cy.wait(2000);
35        cy.get('#root > div > form > div:nth-child(10) > button').click(); // Ajusta el texto según tu aplicación
36        cy.wait(2000);
37
38        //Ingreso historico
39        cy.get(':nth-child(2) > .d-flex > .img-fluid').click();
40        // Esperar 1 segundos
41        cy.wait(1000);
42        //Regresar rutina
```





3. **Pruebas de APIS con POSTMAN**: Postman es una herramienta poderosa para realizar pruebas automáticas de API, permitiendo a los desarrolladores y testers interactuar con sus servicios de manera eficiente.

3.1. Prueba de api para USERS:

GET localhost:5000/api/users/all

Params Authorization Headers (7) Body Scripts Settings

Pre-request

Post-response

```
pm.test("Estado es 200", function () {
    pm.response.to.have.status(200);
});

pm.test("Respuesta contiene una lista de usuarios", function () {
    const jsonData = pm.response.json();
    pm.expect(jsonData).to.be.an("array");
});

pm.test("Cada usuario tiene las propiedades esperadas", function () {
    const jsonData = pm.response.json();
    jsonData.forEach(user => {
        pm.expect(user).to.have.property("_id");
        pm.expect(user).to.have.property("username");
        pm.expect(user).to.have.property("email");
        pm.expect(user).to.have.property("edad");
        pm.expect(user).to.have.property("estatura");
        pm.expect(user).to.have.property("peso");
    });
});
```

Body Cookies Headers (7) Test Results (3/3)

Testing your API endpoints one by one? Run your entire collection and test end-to-end workflows effortlessly. Learn more

Filter Results

PASSED Estado es 200

PASSED Respuesta contiene una lista de usuarios

PASSED Cada usuario tiene las propiedades esperadas

3.2. Pruebas de api para EJERCICIOS:

GET localhost:5000/api/ejercicios/

Params Authorization Headers (7) Body Scripts Settings

Pre-request

Post-response

```
pm.test("Estado es 200", function () {
    pm.response.to.have.status(200);
});

pm.test("La respuesta contiene una lista de ejercicios", function () {
    const jsonData = pm.response.json();
    pm.expect(jsonData).to.be.an("array");
});

pm.test("Cada ejercicio tiene las propiedades esperadas", function () {
    const jsonData = pm.response.json();
    jsonData.forEach(ejercicio => {
        pm.expect(ejercicio).to.have.property("_id");
        pm.expect(ejercicio).to.have.property("nombre");
        pm.expect(ejercicio).to.have.property("objetivo");
        pm.expect(ejercicio).to.have.property("tipo");
        pm.expect(ejercicio).to.have.property("link");
    });
});
```

Body Cookies Headers (7) Test Results (3/3)

Testing your API endpoints one by one? Run your entire collection and test end-to-end workflows effortlessly. Learn more

Filter Results

PASSED Estado es 200

PASSED La respuesta contiene una lista de ejercicios

PASSED Cada ejercicio tiene las propiedades esperadas

3.3 Pruebas de api para RUTINAS:

GET localhost:5000/api/rutinas/672210d94d303ceb9ce74427

Params Authorization Headers (7) Body Scripts Settings

Pre-request

Post-response

```
    pm.response.to.have.status(200);
});

pm.test("La respuesta debe ser un objeto JSON", function () {
    pm.response.to.be.json;
    const jsonData = pm.response.json();
    pm.expect(jsonData).to.be.an("object");
});

pm.test("El objeto rutina debe contener los campos esperados", function () {
    const jsonData = pm.response.json();
    pm.expect(jsonData).to.have.property("_id");
    pm.expect(jsonData).to.have.property("nombre");
    pm.expect(jsonData).to.have.property("createdAt");
});

pm.test("Debería devolver un estado 404 si el ID no es válido", function () {
    if (pm.response.code === 404) {
        const jsonData = pm.response.json();
        pm.expect(jsonData).to.have.property("error", "No existe esa rutina");
    }
});

pm.test("Debería devolver un estado 404 si la rutina no existe", function () {
    if (pm.response.code === 404) {
        const jsonData = pm.response.json();
        pm.expect(jsonData).to.have.property("error", "No existe esa rutina");
    }
```

Body Cookies Headers (7) Test Results (5/5)

Filter Results

PASSED Debería devolver un estado 200 si la rutina existe

PASSED La respuesta debe ser un objeto JSON

PASSED El objeto rutina debe contener los campos esperados

PASSED Debería devolver un estado 404 si el ID no es válido

PASSED Debería devolver un estado 404 si la rutina no existe

CONCLUSIONES

1. El proyecto TrainUp, enfocado en la creación de una aplicación de entrenamiento personal para el gimnasio Go Fitness Gym, alcanzó sus objetivos al implementar una solución tecnológica que optimizó significativamente los procesos operativos. La aplicación transformó la gestión de rutinas, la interacción entre entrenadores y clientes, y el monitoreo del progreso físico de los usuarios. Esta iniciativa se desarrolló en alineación con la norma ISO/IEC 25010, asegurando un enfoque integral en la calidad del software
2. El desarrollo permitió digitalizar procesos esenciales como la personalización de rutinas, la gestión de reservas para clases grupales y el registro de progreso, logrando reducir en un 20% el margen de error y aumentando la eficiencia operativa. Estos avances se reflejaron en un incremento del 25% en la satisfacción de los clientes, medido a través de encuestas de retroalimentación
3. Se adoptó el stack MERN (MongoDB, Express.js, React.js, Node.js) junto con Docker para garantizar una arquitectura escalable y modular. Esta elección tecnológica, respaldada por un análisis de factibilidad técnica y económica, permitió integrar herramientas avanzadas de análisis y visualización de datos, mejorando la experiencia de usuario en un 30%.
4. El sistema implementado cumplió con todos los requerimientos funcionales establecidos, incluyendo el registro de usuarios, la personalización de rutinas y la gestión de notificaciones. Además, destacó en aspectos no funcionales como la seguridad, al implementar encriptación de datos sensibles, y la usabilidad, al diseñar una interfaz intuitiva que facilitó la adopción del sistema por parte de entrenadores y clientes.
5. Las funcionalidades orientadas a la motivación, como gráficos de progreso, recordatorios personalizados y retos semanales, incrementaron la retención de usuarios en un 20%.

Además, el acceso a métricas de rendimiento en tiempo real mejoró la percepción del valor del servicio, reforzando la relación entre los usuarios y el gimnasio.

RECOMENDACIONES

1. **Capacitación Continua en Tecnologías Emergentes:**

   Es fundamental seguir capacitando al equipo en tecnologías emergentes como Docker y el stack MERN (MongoDB, Express.js, React.js, Node.js). Esto asegurará que el equipo esté actualizado y preparado para futuros proyectos, reduciendo tiempos innecesarios y minimizando errores.

2. **Implementación de un Sistema de Monitoreo:**

   Se recomienda implementar un sistema de monitoreo para identificar posibles fallos y áreas de mejora en la aplicación. Esto garantizará un rendimiento óptimo y permitirá una rápida respuesta ante cualquier problema que pueda surgir.

3. **Integración de Retroalimentación Continua:**

   Priorizar la integración de retroalimentación continua de los usuarios es crucial para ajustar las funcionalidades de la aplicación a las necesidades dinámicas del mercado. Esto podría incluir encuestas periódicas y análisis de métricas de uso para identificar áreas de mejora tanto en la interfaz como en la experiencia general.

4. **Auditorías Regulares de Seguridad:**

   Realizar auditorías regulares para garantizar la protección de los datos de los usuarios es esencial. Estas auditorías deben asegurar el cumplimiento con normativas locales e internacionales de privacidad y seguridad digital.

5. **Ampliación de la Compatibilidad de la Aplicación:**

Se recomienda ampliar el alcance de la aplicación para incluir compatibilidad con dispositivos adicionales como tablets y otros sistemas operativos. Esto maximizará el acceso a una mayor base de usuarios y mejorará la versatilidad de la aplicación.

6. **Implementación de un Sistema de Análisis de Uso:**

Implementar un sistema de análisis, como un mapa de calor, para rastrear cómo interactúan los usuarios con la aplicación y medir su nivel de satisfacción. Esto ayudará a identificar mejoras necesarias y nuevas oportunidades de desarrollo.

7. **Mejora del Diseño de la Interfaz:**

Realizar un diseño más interactivo y sencillo de usar para mejorar la experiencia del usuario. Un diseño intuitivo y atractivo puede aumentar la satisfacción y el compromiso de los usuarios con la aplicación.

ANEXOS

Anexo 01. Manual Técnico

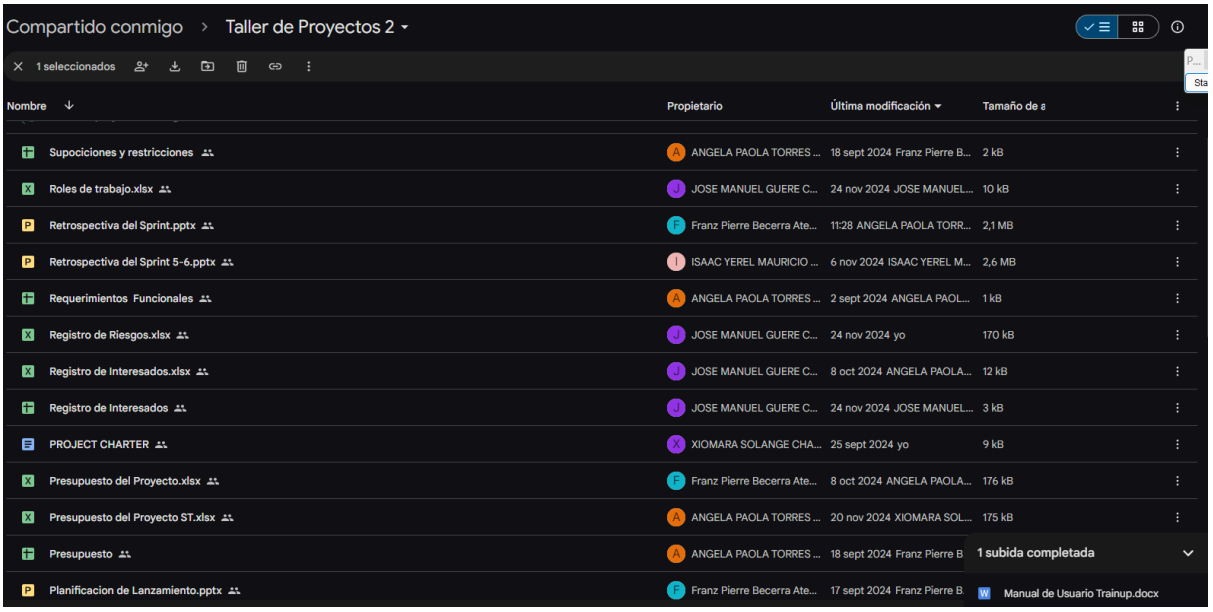
Manual_Tecnico_Trainup.docx

Anexo 02. Manual de Usuario

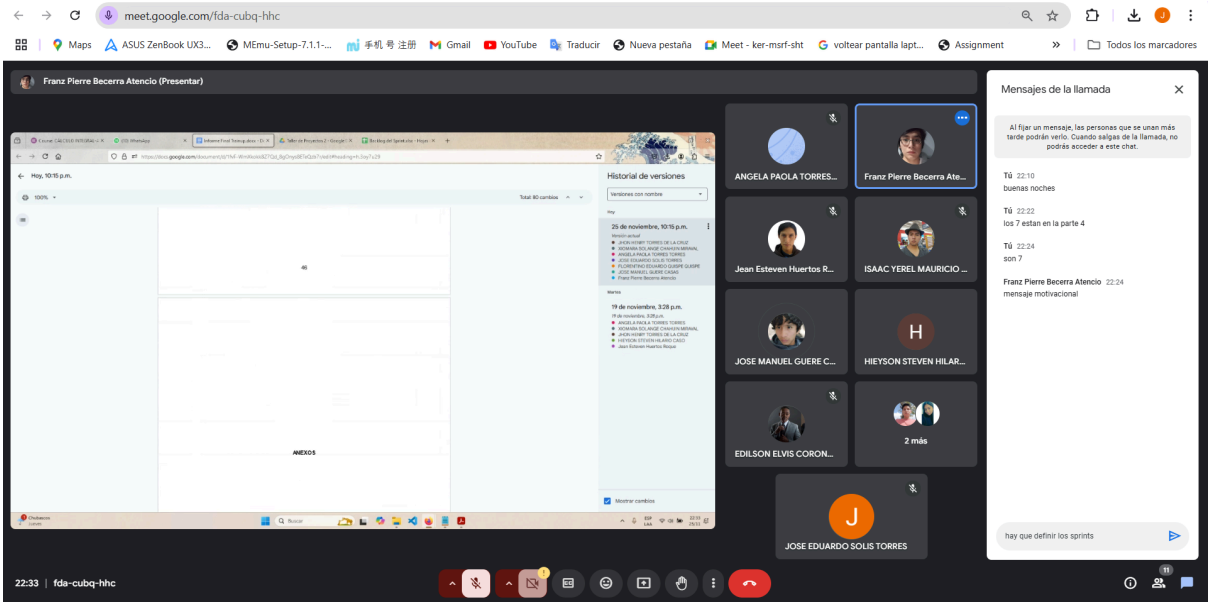
Manual de Usuario Trainup.docx

Anexo 3: Carpeta De Trabajo

Taller de Proyectos 2



Anexo 4: Reuniones



Anexo 5: Video de demostración de la aplicación web

https://youtu.be/MKLzaGMuOoM?si=pRX3lvJgvmqbBDDF

Anexo 6: Presentación Final del proyecto

https://www.canva.com/design/DAGYLU6geWQ/he2_W2w_axFgsq27k_lGqw/edit?utm_content=DAGYLU6geWQ&utm_campaign=designshare&utm_medium=link2&utm_source=sharebutton