

METAFIT

NOMBRES DE LOS ESTUDIANTES

EDGAR ANDRES LEAL PLATA

KAROLAIN HUÉRFANO REYES

DOCENTES INVOLUCRADOS

JUAN CARLOS MARIÑO MORANTES

**CAMPUSLANDS
RUTA NODE JS
FLORIDABLANCA
2025**

METAFIT

1. SITUACIÓN PROBLEMA

La administración de un gimnasio o de los clientes de un entrenador personal suele implicar múltiples procesos —registro de usuarios, seguimiento físico, control de planes de entrenamiento, nutrición y manejo de pagos— que, cuando se realizan en hojas de cálculo o de manera manual, generan **pérdida de tiempo, errores de registro y poca visibilidad del progreso y las finanzas**.

Esta situación crea la necesidad de contar con una **herramienta unificada, simple y confiable**, que permita **gestionar de forma integral** a los clientes, sus rutinas, contratos y movimientos financieros, garantizando además la **consistencia de los datos**.

El proyecto busca aprovechar esta oportunidad desarrollando una **aplicación de línea de comandos (CLI) en Node.js**, que automatice estas tareas, mejore el control del negocio y facilite el seguimiento del rendimiento de cada cliente y de la operación del gimnasio.

2. LEVANTAMIENTO DE REQUERIMIENTOS

El levantamiento de requerimientos consistió en recopilar, analizar y priorizar las necesidades del proyecto mediante reuniones de definición y revisión entre los integrantes.

Se identificaron las funcionalidades principales, los flujos de usuario y las restricciones técnicas (uso de Node.js, MongoDB y CLI), estableciendo un backlog inicial en el tablero Kanban.

Este proceso permitió obtener una visión clara del alcance, asegurar que las expectativas del producto fueran compartidas por todos y sentar las bases para el desarrollo ágil.



3. REQUERIMIENTOS

3.1. Requerimientos Funcionales

TIPO	DESCRIPCIÓN	PRIORIDAD
FUNCIONALES	CRUD de clientes con validaciones	ALTA
	CRUD de planes de entrenamiento con contratos automáticos	ALTA
	Registro de avances físicos y visualización cronológica	MEDIA
	Gestión de planes de nutrición y reportes semanales	MEDIA
	Registro de ingresos y egresos con balance financiero	ALTA

3.2. Requerimientos No Funcionales

TIPO	DESCRIPCIÓN	PRIORIDAD
NO FUNCIONALES	Persistencia en MongoDB con transacciones reales	ALTA
	Validaciones de datos en 'models/'	ALTA
	Interfaz CLI amigable usando 'chalk' e 'inquirer'	MEDIA
	Código POO, SOLID y patrones de diseño	MEDIA
	Rollback automático en cancelaciones críticas	ALTA

3.3 Requerimientos Especiales

TIPO	DESCRIPCIÓN	PRIORIDAD
ESPECIALES	Contratos generados automáticamente al asignar un plan	ALTA
	Reporte financiero filtrado por fecha/cliente	MEDIA

4. HISTORIAS DE USUARIO CON CRITERIOS DE ACEPTACIÓN

1.

HISTORIA DE USUARIO			
Prioridad: Alta			
CÓDIGO DEL REQUERIMIENTO:	RF01	Actor	Administrador / Entrenador
NOMBRE DEL REQUERIMIENTO	Gestión de clientes		
Descripción			
Como administrador o entrenador puedo registrar, actualizar, eliminar y listar clientes para mantener actualizada la base de datos y poder asociarlos a planes de entrenamiento, contratos y seguimientos.			
Funcionalidad			
El sistema permitirá crear un registro de cliente con datos personales (nombre, contacto, estado del plan), editar información existente, eliminar registros y mostrar una lista de clientes activos o inactivos.			
Criterios de aceptación	1. El sistema debe permitir crear un cliente con campos obligatorios (nombre, contacto, fecha de ingreso). 2. El sistema debe permitir actualizar la información de un cliente existente. 3. El sistema debe permitir eliminar un cliente solo si no tiene planes activos o con rollback en caso de error. 4. El sistema debe mostrar en consola la lista de clientes registrados.		
Restricciones			
Los datos deben validarse en el modelo para evitar registros incompletos o con formato incorrecto.			

2.

HISTORIA DE USUARIO			
Prioridad: Alta			
CÓDIGO DEL REQUERIMIENTO:	RF02	Actor	Administrador/ Entrenador
NOMBRE DEL REQUERIMIENTO	Gestión de planes de entrenamiento		
Descripción			
Como administrador o entrenador puedo crear, actualizar, asociar y finalizar planes de entrenamiento para asignarlos a los clientes y dar seguimiento a su progreso.			
Funcionalidad			
El sistema permitirá definir un plan con nombre, duración, metas físicas y nivel (principiante, intermedio, avanzado); asociarlo a uno o varios clientes; y registrar renovaciones, cancelaciones o finalización.			
Criterios de aceptación	1.El sistema debe permitir crear un plan con todos los campos requeridos. 2. El sistema debe permitir asociar un plan a uno o varios clientes existentes. 3.El sistema debe generar automáticamente un contrato al asociar un plan. 4. El sistema debe permitir cancelar o finalizar un plan, aplicando rollback en seguimientos si corresponde.		
Restricciones			
No se puede asignar un plan a un cliente inexistente o sin datos completos.			

3.

HISTORIA DE USUARIO			
Prioridad: Media			
CÓDIGO DEL REQUERIMIENTO:	RF03	Actor	Entrenador
NOMBRE DEL REQUERIMIENTO	Seguimiento fisico del cliente		
Descripción			
Como entrenador puedo registrar avances semanales (peso, grasa corporal, medidas, fotos, comentarios) para evaluar la evolución física de cada cliente.			
Funcionalidad			
El sistema permitirá crear registros cronológicos de avances, visualizarlos en orden de fecha y eliminar registros cuando sea necesario, garantizando consistencia con el plan activo.			
Criterios de aceptación	1. El sistema debe permitir agregar un avance con fecha, peso, grasa corporal y comentarios. 2. El sistema debe permitir listar el historial de avances de un cliente en orden cronológico. 3. El sistema debe permitir eliminar un registro, aplicando rollback si afecta la coherencia del plan.		
Restricciones			
Solo se pueden registrar avances de clientes con un plan activo.			



4.

HISTORIA DE USUARIO			
Prioridad: Media			
CÓDIGO DEL REQUERIMIENTO:	RF04	Actor	Nutricionista/ Entrenador
NOMBRE DEL REQUERIMIENTO	Gestión de Planes de Nutrición		
Descripción			
Como nutricionista o entrenador puedo crear y asociar planes de alimentación a cada cliente para complementar su entrenamiento.			
Funcionalidad			
El sistema permitirá definir alimentos por día, calorías estimadas y generar un reporte semanal			
Criterios de aceptación	1. El sistema debe permitir crear un plan nutricional vinculado a un cliente y a su plan de entrenamiento. 2. El sistema debe permitir registrar alimentos con calorías diarias. 3. El sistema debe generar un reporte nutricional semanal.		
Restricciones			
No se puede crear un plan de nutrición si el cliente no tiene un plan de entrenamiento activo.			

5.

HISTORIA DE USUARIO			
Prioridad: Alta			
CÓDIGO DEL REQUERIMIENTO:	RF05	Actor	Administrador
NOMBRE DEL REQUERIMIENTO	Gestión financiera con transacciones		
Descripción			
Como administrador puedo registrar ingresos y egresos para llevar control del balance financiero y asegurar la consistencia en pagos.			
Funcionalidad			
El sistema permitirá ingresar pagos de mensualidades, sesiones y egresos de servicios; realizar consultas por fecha o cliente; y manejar transacciones para evitar inconsistencias en caso de error.			
Criterios de aceptación	1. El sistema debe permitir registrar ingresos y egresos con fecha, monto y descripción. 2. El sistema debe permitir consultar el balance por rango de fechas o clientes. 3. Las operaciones de pago deben realizarse en transacciones reales en Mongo DB.		
Restricciones			
Las transacciones deben confirmarse o revertirse por completo si ocurre una falla.			



6.

HISTORIA DE USUARIO			
Prioridad: Alta			
CÓDIGO DEL REQUERIMIENTO:	RF06	Actor	Administrador
NOMBRE DEL REQUERIMIENTO	Autenticación de usuario (Login simulado)		
Descripción			
Como administrador puedo iniciar sesión en el sistema para acceder de forma segura a las funciones de gestión de clientes, plan, contratos y finanzas.			
Funcionalidad			
El sistema permitirá que el administrador ingrese un usuario y contraseña simulados para validar el acceso. Al iniciar sesión, se mostrará un saludo personalizado y habilitarán las funciones del sistema.			
Criterios de aceptación	1. El sistema debe solicitar usuario y contraseña en la consola. 2. El sistema debe validar los datos y permitir el acceso solo si son correctos. 3. Al iniciar sesión, el sistema debe mostrar un mensaje de bienvenida con el nombre del usuario.		
Restricciones			
El login es simulado; las credenciales estarán almacenadas en variables de entorno (.env) y no habrá múltiples roles.			

7.

HISTORIA DE USUARIO			
Prioridad: Alta			
CÓDIGO DEL REQUERIMIENTO:	RF07	Actor	Administrador
NOMBRE DEL REQUERIMIENTO	Generación automática de contratos.		
Descripción			
Como administrador puedo generar un contrato automáticamente cada vez que asigna un plan de entrenamiento a un cliente para formalizar el acuerdo.			
Funcionalidad			
El sistema generará un contrato con los campos: condiciones, duración, precio, fecha de inicio y fin, y lo asociará al cliente y al plan. El contrato quedará guardado en la base de datos para futuras consultas.			
Criterios de aceptación	1. El sistema debe generar el contrato en el momento de asociar un plan a un cliente. 2. El contrato debe incluir todos los campos obligatorios y guardar la relación con el cliente y el plan. 3. El sistema debe permitir consultar los contratos registrados por el cliente o plan.		
Restricciones			
No se puede generar un contrato si el cliente no está registrado o si el plan no existe.			

8.

HISTORIA DE USUARIO			
Prioridad: Alta			
CÓDIGO DEL REQUERIMIENTO:	RF08	Actor	Administrador
NOMBRE DEL REQUERIMIENTO	Reportes de progreso y finanzas		
Descripción			
Como administrador puedo generar reportes de progreso físico y de balance financiero para evaluar el desempeño de los clientes y del gimnasio.			
Funcionalidad			
El sistema permitirá generar reportes en consola con datos resumidos de avances físicos por cliente y balances financieros por rango de fechas.			
Criterios de aceptación	1. El sistema debe permitir seleccionar el tipo de reporte (progreso físico o financiero). 2. El reporte de progreso debe mostrar avances cronológicos del cliente elegido. 3. El reporte financiero debe mostrar ingresos, egresos y balance en el rango de fechas solicitado.		
Restricciones			
Los reportes sólo pueden generarse si existen datos registrados en la base de datos.			

9.

HISTORIA DE USUARIO			
Prioridad: Alta			
CÓDIGO DEL REQUERIMIENTO:	RF09	Actor	Administrador / Entrenador
NOMBRE DEL REQUERIMIENTO	Cancelación de plan con Rollback		
Descripción			
Como administrador o entrenador puedo cancelar un plan de entrenamiento de un cliente y el sistema debe revertir automáticamente los seguimientos y pagos asociados para mantener la consistencia de datos.			
Funcionalidad			
El sistema permitirá cancelar un plan activo, eliminar los registros de seguimiento físico y revertir los pagos asociados mediante una transacción en MongoDB que asegure el rollback en caso de error.			
Criterios de aceptación	1. El sistema debe permitir seleccionar el cliente y el plan a cancelar. 2. Al cancelar, el sistema debe eliminar o revertir seguimientos, pagos y el contrato asociado. 3. En caso de fallo, la operación debe deshacerse por completo para no dejar datos inconsistentes.		
Restricciones			
Solo se pueden cancelar planes que estén activos; no se puede cancelar un plan ya finalizado.			



5. METODOLOGÍA

Para el desarrollo del proyecto se adoptó la metodología ágil Kanban, la cual permite gestionar el trabajo de manera visual y flexible.

Kanban se basa en un tablero dividido en columnas (*Backlog*, *Por hacer*, *En progreso*, *En revisión* y *Hecho*), donde cada tarjeta representa una historia de usuario o tarea.

Este enfoque facilita el seguimiento continuo, la priorización dinámica y la entrega incremental, optimizando el flujo de trabajo sin depender de iteraciones largas.

5.1 Orden de las actividades

El proyecto se ejecutará en **flujos de trabajo progresivos** (waves) para asegurar que las funcionalidades críticas estén listas primero:

1. Preparación (Sprint Planning)

- Configuración del entorno de desarrollo (Node.js, MongoDB, repositorio Git).
- Creación del tablero Kanban y priorización de historias de usuario.
- Definición de estructura de carpetas y dependencias npm.

2. Desarrollo del Core (Sprint Execution)

- Implementación de CRUD de clientes y planes de entrenamiento.
- Generación automática de contratos con transacciones en MongoDB.
- Registro de ingresos y egresos financieros.

3. Módulos de Seguimiento y Nutrición (Sprint Review)

- Registro de avances físicos y reportes cronológicos.
- Creación de planes de alimentación y reportes de calorías.

4. Cierre y Optimización (Sprint Retrospective)

- Implementación de rollback en cancelación de planes.
- Pruebas integrales, ajustes de UX en CLI y documentación final.

Durante todo el proceso, las tareas se moverán en el tablero Kanban conforme avancen de *Por hacer* → *En progreso* → *En revisión* → *Hecho*, manteniendo la **entrega continua** y permitiendo ajustes en tiempo real.



5.2 Roles y responsabilidades

ROL	INTEGRANTE	RESPONSABILIDADES
Product Owner (PO)	Karolain Reyes	Define la visión del producto, prioriza el backlog, valida historias de usuario, revisa entregables y asegura que se cumplan los objetivos de negocio.
Scrum Master (SM)	Andres Leal	Facilita la comunicación, elimina bloqueos, vela por el cumplimiento de los principios ágiles y asegura que el equipo mantenga un flujo constante en Kanban.
Developers (DEV)	Karolain Reyes / Andres Leal	Implementan las funcionalidades, realizan pruebas, documentan el código y garantizan la calidad técnica (POO, SOLID, transacciones MongoDB).

6. EVIDENCIA DE PLANTEAMIENTO DE PLATAFORMA DE TRABAJO

Acá se debe documentar toda la evidencia de trabajo colaborativo con los siguientes elementos:

- Link del repositorio donde se evidencia el trabajo colaborativo con el correcto uso de roles, ramas y conventional commit.

[Repositorio Github](#)

- Link de los videos que se grabaron en las reuniones realizadas (Sprint planning, Daily Stand Up, Sprint Retrospective)
 - [Sprint Planning](#)
 - [Sprint Execution](#)
 - [Sprint Review](#)
 - [Sprint Retrospective](#)
 - [Daily Standup](#)

- Evidencia (capturas) del tablero Scrum utilizado en todas las etapas donde se visualicen todas las tareas, requerimientos e historias de usuario documentadas allí con responsables, tiempos, priorización, y demás requerimientos que se indiquen para las tareas.

DIVISIÓN DE SPRINTS Y OBJETIVOS GENERALES DEL PROYECTO

Todas las reuniones	
Reunión	Objetivo
 Sprint Planning  1	 Objetivo: Tener lista la base del proyecto y el tablero de trabajo.
 Sprint Execution  1	 Objetivo: Construir las funcionalidades críticas de negocio.
 Sprint Review	 Objetivo: Añadir seguimiento físico y nutrición para enriquecer el producto.
 Sprint Retrospective	 Objetivo: Estabilizar, optimizar y documentar.

REQUERIMIENTOS:

Requerimientos del producto.

Tipo	Descripción	Prioridad
Funcional	CRUD de clientes con validaciones.	★ ★ ★
	CRUD de planes de entrenamiento con contratos automáticos.	★ ★ ★
	Registro de avances físicos y visualización cronológica.	★ ★
	Gestión de planes de nutrición y reportes semanales.	★ ★
	Registro de ingresos y egresos con balance financiero.	★ ★ ★
No Funcional	Persistencia en MongoDB con transacciones reales .	★ ★ ★
	Validaciones de datos en <code>/models</code> .	★ ★ ★
	Interfaz CLI amigable usando <code>chalk</code> , <code>inquirer</code> .	★ ★
	Código con POO, SOLID y dos patrones de diseño (ej. <code>Repository</code> , <code>Command</code>).	★ ★
	Rollback automático en cancelaciones críticas.	★ ★ ★
Especiales	Contratos generados automáticamente al asignar un plan.	★ ★ ★
	Reporte financiero filtrado por fecha/cliente.	★ ★

EJEMPLO DE HISTORIAS DE USUARIO:

Historias de Usuario

Las historias están priorizadas por impacto en el negocio (debe → debería → podría).

● [HU-CLI-01] CRUD de clientes

Como entrenador

quiero crear, listar, actualizar y eliminar clientes

para mantener la información actualizada.

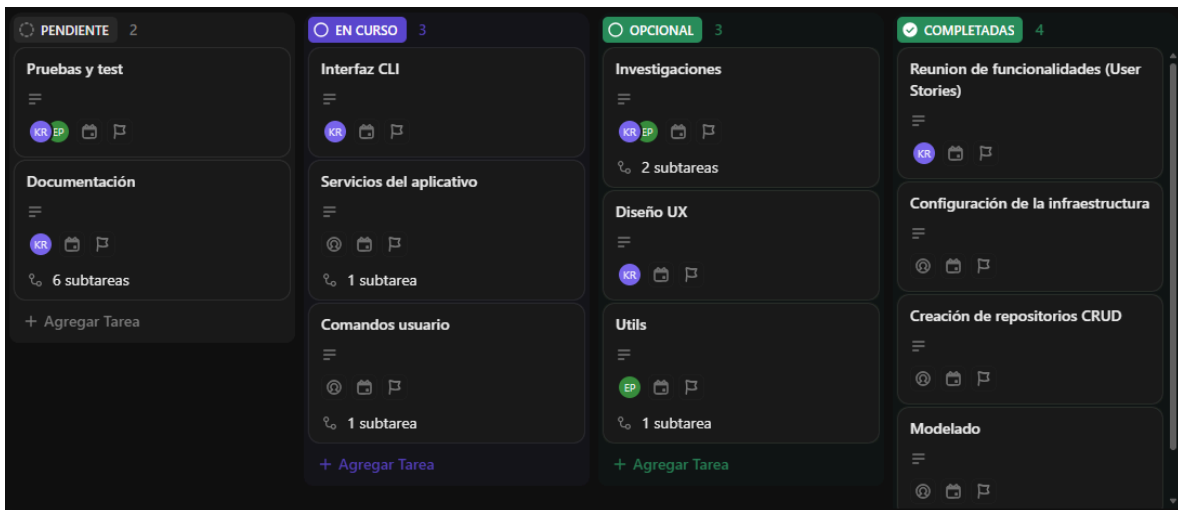
Checklist – Criterios de aceptación

- ☐ Validar campos: nombre, edad, contacto.
- ☐ Listado con paginación en consola.
- ☐ Confirmación antes de eliminar.

Para ver más información sobre la organización de las historias de usuario y los requerimientos puedes ver la organización en nuestro [Notion](#).



TABLERO KANBAN



Este tablero representa un progreso de nuestro proyecto, para ver el tablero completo y la lista del backlog puedes dar click [aquí](#).

- Documentación de los resultados y funcionamiento del producto final resaltando tecnologías utilizadas y cumplimiento de trabajo basado en las tareas planteadas.

A continuación una capturas que demuestran algunas de las vistas del proyecto

SALUDO BIENVENIDA



MOVIMIENTOS

```
✓ Elige una opción: Listar clientes
✓ Elige una opción: Ingreso
===== Movimientos por Tipo =====
-----
29/09/2025, 19:20
María Fernanda López
Plan Quema Grasa Express
150000
-----
29/09/2025, 19:20
María Fernanda López
Plan Quema Grasa Express
150000
-----
29/09/2025, 20:49
Felipe Montañez
Plan Hipertrofia Total
250000
-----
? Presiona cualquier tecla para volver
```

BALANCE GENERAL

```
✓ Elige una opción: Listar clientes
===== Balance =====
Ingresos: 550000
Egresos: 13500
Balance: 536500
? Presione cualquier boton para volver
```

CLIENTES ALMACENADOS

Nombre: Sebastián Mauricio Ríos
Documento: 1010987654
Telefono: 3089012345
Planes:

Nombre: Valentina Isabel Mejía
Documento: 1009876543
Telefono: 3090123456
Planes:

Nombre: Juancho
Documento: 1090300300
Telefono: 3101001001
Planes:

Nombre: Edgar Andres Leal
Documento: 1092175439
Telefono: 3223578559
Planes:

Nombre: Felipe Montañez
Documento: 1000000000
Telefono: 3103103100
Planes: 68db20a1da2c95e6d424bfe3

Nombre: Maria Becerra
Documento: 1090100100
Telefono: 3103003000
Planes: 68db20a1da2c95e6d424bfe7

=====

? Presione cualquier tecla para volver



7. CONCLUSIONES

Conclusión

El desarrollo de esta herramienta CLI permitió crear una solución **ágil, escalable y eficiente** para la gestión integral de un gimnasio o entrenador personal.

Gracias a la aplicación de **JavaScript, Node.js y MongoDB**, junto con principios de **Programación Orientada a Objetos, SOLID** y transacciones reales, se logró un sistema confiable para manejar clientes, planes de entrenamiento, seguimiento físico, nutrición y finanzas desde la terminal.

La utilización de la metodología ágil **Kanban**, apoyada en **ClickUp y Notion**, facilitó una planificación clara, priorización de tareas y un seguimiento continuo del progreso, garantizando que las funcionalidades críticas se desarrollaran en el tiempo establecido.

Este proyecto demuestra que, incluso con un equipo pequeño, es posible construir herramientas profesionales que combinan **simplicidad, control y robustez**, brindando a los entrenadores y gimnasios una forma práctica de optimizar su trabajo y enfocarse en lo más importante: **el bienestar y progreso de sus clientes**.

Conclusión – Sprint Retrospective

Durante este sprint logramos completar las funcionalidades planificadas y mantener un flujo de trabajo constante gracias al uso de **Kanban**.

El equipo destacó una buena **comunicación** y una distribución equilibrada de las tareas, lo que permitió resolver imprevistos sin retrasos.

Identificamos oportunidades de mejora en la **documentación de cambios** y en la estimación de tiempos, aspectos que ajustaremos para el próximo ciclo.

En general, el sprint demostró que nuestra metodología y herramientas son adecuadas para mantener un desarrollo **ágil, organizado y enfocado en los objetivos del proyecto**.

