

**GIMNASIO LOS SAYAYIN**

**NOMBRES DE LOS ESTUDIANTES**

- BRYAN ANDRES VILLABONA ALMEYDA
- JOAN SEBASTIAN GOMEZ SERRANO
- SERGIO STEVEN LIEVANO AMAYA

**GRUPO ASIGNADO**

**GRUPO 5**

**DOCENTES INVOLUCRADOS**

- JUAN CARLOS MARIÑO MORANTES

**CAMPUSLANDS  
SALON U1  
RUTA NODE.JS  
FLORIDABLANCA  
2025**

## ❖ SITUACIÓN PROBLEMA

El cliente Gimnasio Los Sayayines enfrenta desafíos significativos en la gestión de sus operaciones diarias. Específicamente, se han identificado problemas en las siguientes áreas clave: gestión de clientes, seguimiento de ventas, administración de planes de membresía, monitoreo del progreso físico de los usuarios, gestión de contratos y control financiero. Estas deficiencias están afectando la eficiencia operativa y la capacidad del gimnasio para ofrecer un servicio óptimo a sus miembros.

Para abordar estas necesidades, se propone el diseño y desarrollo de una aplicación robusta utilizando Node.js para el backend y una base de datos MongoDB para el almacenamiento de datos. Esta solución tecnológica permitirá centralizar y optimizar la gestión de todos los aspectos mencionados, proporcionando herramientas eficientes para el seguimiento de clientes, automatización de ventas y planes, registro de seguimiento físico, administración de contratos y una gestión financiera integral.

## ❖ LEVANTAMIENTO DE REQUERIMIENTOS

Para el desarrollo del aplicativo del Gimnasio Los Sayayines, se realizará un levantamiento exhaustivo de requerimientos funcionales y no funcionales, con el objetivo de asegurar que la solución tecnológica propuesta aborde de manera integral los desafíos identificados. Este proceso incluirá la interacción con los stakeholders clave del gimnasio para comprender a fondo sus necesidades y expectativas.

Los requerimientos se centrarán en las siguientes áreas, basándose en la situación problema descrita:

- ❖ Gestión de Clientes:
  - Registro y almacenamiento de información detallada de los clientes.
  - Funcionalidades para la búsqueda, edición y eliminación de perfiles de clientes.
  - Asignación de planes de membresía a los clientes.
- ❖ Seguimiento de Ventas:
  - Registro de todas las transacciones de ventas.
  - Generación de reportes de ventas por período, tipo de producto/servicio, etc.
  - Gestión de pagos y facturación.
- ❖ Administración de Planes de Membresía:
  - Creación, edición y gestión de diferentes tipos de planes de membresía.
  - Control de la activación y expiración de las membresías.
  - Notificaciones automáticas sobre la renovación de planes.
- ❖ Monitoreo del Progreso Físico de los Usuarios:
  - Registro de datos de seguimiento físico.
  - Visualización del progreso a lo largo del tiempo para los usuarios y entrenadores.
  - Posibilidad de adjuntar planes de entrenamiento personalizados.
- ❖ Gestión de Contratos:
  - Creación y almacenamiento digital de contratos de membresía.
  - Seguimiento del estado de los contratos.
  - Generación de documentos contractuales.
- ❖ Control Financiero:
  - Registro de ingresos y egresos.

- Generación de estados financieros y reportes de rentabilidad.
- Integración con módulos de ventas y membresías para una gestión financiera unificada.

La metodología de levantamiento de requerimientos incluirá técnicas como entrevistas, encuestas, observación y análisis de documentos existentes, asegurando una comprensión completa de los procesos actuales y las necesidades futuras del Gimnasio Los Sayayines. Los requerimientos serán documentados de forma clara y detallada, sirviendo como base para el diseño y desarrollo del aplicativo en Node.js y MongoDB.

## ❖ **REQUERIMIENTOS**

### ➤ **Requerimientos Funcionales**

- ❖ **Gestión de Clientes:**
  - Crear clientes.
  - Listar clientes.
  - Actualizar clientes.
  - Eliminar clientes.
  - Asociar clientes a uno o varios planes de entrenamiento.
- ❖ **Gestión de Planes de Entrenamiento:**
  - Crear planes con nombre, duración, metas físicas y nivel (principiante, intermedio, avanzado).
  - Asociar planes a uno o varios clientes.
  - Registrar contrato al asignar el plan a un cliente.
  - Permitir renovar un plan.
  - Permitir cancelar un plan.
  - Permitir finalizar un plan.
- ❖ **Seguimiento Físico:**
  - Registrar avances semanales (peso, grasa corporal, medidas, fotos, comentarios).
  - Visualizar progreso cronológico.
  - Permitir eliminar registros de seguimiento físico.
- ❖ **Nutrición:**
  - Crear planes de alimentación asociados al cliente y al plan de entrenamiento.
  - Registrar alimentos por día (con calorías estimadas).
  - Consultar reporte nutricional semanal.
- ❖ **Contratos:**
  - Generar contratos automáticamente al asignar un plan.
  - Visualizar campos del contrato: condiciones, duración, precio, fecha inicio/fin.
  - Asociar el contrato al cliente y plan correspondiente.
- ❖ **Gestión Financiera:**
  - Registrar ingresos (mensualidades, sesiones individuales).
  - Registrar egresos (servicios, suplementos, gastos operativos).
  - Consultar balance financiero por fecha o por cliente.

➤ **Requerimientos No Funcionales.**

- ❖ Seguimiento Físico:
  - Implementación de rollback si la eliminación de registros afecta la consistencia del plan.
- ❖ Gestión Financiera:
  - Implementación de transacciones para evitar inconsistencias en pagos.

❖ **HISTORIAS DE USUARIO CON CRITERIOS DE ACEPTACIÓN**

1.

HISTORIA DE USUARIO			
Prioridad: Alta			
CÓDIGO DEL REQUERIMIENTO:	RF01	Actor	Administrador
NOMBRE DEL REQUERIMIENTO	Creación de Cliente		
Descripción			
Como Administrador, quiero poder crear un nuevo cliente con sus datos personales (nombre, apellido, email, teléfono, etc.), para registrar a los nuevos miembros del gimnasio.			
Funcionalidad			
Permitir al administrador ingresar y guardar la información básica de un nuevo cliente.			
Criterios de aceptación	1. El sistema debe permitir ingresar todos los datos requeridos (nombre, apellido, email, teléfono). 2. El sistema debe validar que el email sea único 3. El sistema debe guardar el nuevo cliente en la base de datos.		
Restricciones			
El email debe ser un formato válido.			

**2.**

HISTORIA DE USUARIO			
Prioridad: Alta			
CÓDIGO DEL REQUERIMIENTO:	RF02	Actor	Administrador
NOMBRE DEL REQUERIMIENTO	Listado de Clientes.		
Descripción			
Como Administrador, quiero poder listar todos los clientes existentes, para tener una visión general de los miembros del gimnasio.			
Funcionalidad			
Mostrar una lista paginada de todos los clientes registrados con su información principal.			
Criterios de aceptación	1. El listado debe mostrar al menos nombre, apellido y estado del plan. 2. El sistema debe permitir ordenar y filtrar la lista de clientes.		
Restricciones			
Ninguna.			

**3.**

<b>HISTORIA DE USUARIO</b>			
<b>Prioridad:</b> Alta			
<b>CÓDIGO DEL REQUERIMIENTO:</b>	RF03	<b>Actor</b>	Administrador
<b>NOMBRE DEL REQUERIMIENTO</b>	Listado de Clientes.		
<b>Descripción</b>			
Como Administrador, quiero poder listar todos los clientes existentes, para tener una visión general de los miembros del gimnasio.			
<b>Funcionalidad</b>			
Mostrar una lista paginada de todos los clientes registrados con su información principal.			
<b>Criterios de aceptación</b>	1. El listado debe mostrar al menos nombre, apellido y estado del plan. 2. El sistema debe permitir ordenar y filtrar la lista de clientes.		
<b>Restricciones</b>			
Ninguna.			

4.

HISTORIA DE USUARIO			
Prioridad: Alta			
CÓDIGO DEL REQUERIMIENTO:	RF04	Actor	Administrador
NOMBRE DEL REQUERIMIENTO	Eliminación de Cliente.		
Descripción			
Como Administrador, quiero poder eliminar un cliente del sistema, para gestionar los miembros que ya no están activos.			
Funcionalidad			
Permitir al administrador borrar un cliente y toda su información asociada.			
Criterios de aceptación	1.El sistema debe solicitar confirmación antes de eliminar un cliente. 2..El sistema debe eliminar el cliente de la base de datos 3. El sistema debe asegurar que no queden registros huérfanos asociados al cliente eliminado.		
Restricciones			
Ninguna.			

5.

HISTORIA DE USUARIO			
Prioridad: Media.			
CÓDIGO DEL REQUERIMIENTO:	RF05	Actor	Administrador
NOMBRE DEL REQUERIMIENTO	Asociación de Cliente a Plan de Entrenamiento		
Descripción			
Como Administrador, quiero poder asociar un cliente a uno o varios planes de entrenamiento, para asignarles un régimen de ejercicios.			
Funcionalidad			
Permitir al administrador seleccionar un cliente y asignarle uno o más planes de entrenamiento disponibles.			
Criterios de aceptación	1. El sistema debe mostrar una lista de planes de entrenamiento disponibles. 2. El sistema debe permitir seleccionar uno o varios planes para un cliente. 3. El sistema debe registrar la asociación entre el cliente y el plan.		
Restricciones			
Un cliente no puede tener dos planes activos del mismo tipo simultáneamente.			

6.

HISTORIA DE USUARIO			
Prioridad: Alta			
CÓDIGO DEL REQUERIMIENTO:	RF06	Actor	Administrador
NOMBRE DEL REQUERIMIENTO	Creación de Plan de Entrenamiento.		
Descripción			
Como Administrador, quiero poder crear planes de entrenamiento con nombre, duración, metas físicas y nivel (principiante, intermedio, avanzado), para ofrecer diversas opciones a los clientes.			
Funcionalidad			
Permitir al administrador ingresar y guardar los detalles de un nuevo plan de entrenamiento.			
Criterios de aceptación	1. El sistema debe permitir ingresar todos los datos requeridos (nombre, duración, metas físicas, nivel). 2. El sistema debe validar que el nombre del plan sea único. 3.El sistema debe guardar el nuevo plan en la base de datos.		
Restricciones			
La duración del plan debe ser un valor numérico positivo.			

7.

HISTORIA DE USUARIO			
Prioridad: Media.			
CÓDIGO DEL REQUERIMIENTO:	RF07	Actor	Administrador
NOMBRE DEL REQUERIMIENTO	Asociación de Plan a Cliente.		
Descripción			
Como Administrador, quiero poder asociar planes a uno o varios clientes, para organizar los planes de entrenamiento.			
Funcionalidad			
Permitir al administrador seleccionar un plan y asignarlo a uno o más clientes.			
Criterios de aceptación	1. El sistema debe mostrar una lista de clientes disponibles. 2. El sistema debe permitir seleccionar uno o varios clientes para un plan. 3. El sistema debe registrar la asociación entre el plan y el cliente.		
Restricciones			
Ninguna.			

8.

HISTORIA DE USUARIO			
Prioridad: Alta			
CÓDIGO DEL REQUERIMIENTO:	RF08	Actor	Administrador
NOMBRE DEL REQUERIMIENTO	Registro de Contrato al Asignar Plan.		
Descripción			
Como Administrador, quiero que se registre un contrato automáticamente al asignar un plan a un cliente, para formalizar el acuerdo.			
Funcionalidad			
El sistema debe generar y registrar un contrato con los detalles del plan y el cliente al momento de la asignación.			
Criterios de aceptación	1. El sistema debe generar un contrato con los datos del cliente y del plan. 2. El sistema debe guardar el contrato en la base de datos con un estado "activo". 3.El sistema debe mostrar un resumen del contrato al administrador.		
Restricciones			
No se puede asignar un plan sin generar un contrato.			

9.

HISTORIA DE USUARIO			
Prioridad: Alta			
CÓDIGO DEL REQUERIMIENTO:	RF09	Actor	Cliente/Administrador
NOMBRE DEL REQUERIMIENTO	Renovación de Plan		
Descripción			
Como Cliente/Administrador, quiero poder renovar un plan de entrenamiento, para continuar con mi régimen o el de un cliente.			
Funcionalidad			
Permitir la extensión de la duración de un plan de entrenamiento existente.			
Criterios de aceptación	1. El sistema debe permitir seleccionar un plan activo para renovar. 2.El sistema debe actualizar la fecha de finalización del plan. 3.El sistema debe generar una nueva factura o registro de pago por la renovación.		
Restricciones			
Solo se pueden renovar planes activos o próximos a expirar.			



**10.**

<b>HISTORIA DE USUARIO</b>			
<b>Prioridad:</b> Media.			
<b>CÓDIGO DEL REQUERIMIENTO:</b>	RF10	<b>Actor</b>	Cliente/Administrador
<b>NOMBRE DEL REQUERIMIENTO</b>	<b>Cancelación de Plan.</b>		
<b>Descripción</b>			
Como Cliente/Administrador, quiero poder cancelar un plan de entrenamiento, para finalizar el servicio antes de tiempo.			
<b>Funcionalidad</b>			
Permitir la terminación anticipada de un plan de entrenamiento.			
<b>Criterios de aceptación</b>	1. El sistema debe permitir seleccionar un plan activo para cancelar. 2. El sistema debe cambiar el estado del plan a "cancelado". 3.El sistema debe registrar la fecha de cancelación.		
<b>Restricciones</b>			
Puede haber penalizaciones o políticas de reembolso asociadas a la cancelación.			

11.

<b>HISTORIA DE USUARIO</b>			
<b>Prioridad:</b> Media.			
<b>CÓDIGO DEL REQUERIMIENTO:</b>	RF11	<b>Actor</b>	Cliente/Administrador
<b>NOMBRE DEL REQUERIMIENTO</b>	<b>Finalización de Plan.</b>		
<b>Descripción</b>			
Como Cliente/Administrador, quiero poder finalizar un plan de entrenamiento, para marcar su culminación natural.			
<b>Funcionalidad</b>			
Permitir marcar un plan de entrenamiento como completado al finalizar su duración.			
<b>Criterios de aceptación</b>	1. El sistema debe permitir seleccionar un plan cuya duración haya expirado o esté por expirar. 2.El sistema debe cambiar el estado del plan a "finalizado". 3.El sistema debe registrar la fecha de finalización.		
<b>Restricciones</b>			
Un plan solo puede finalizar si ha cumplido su duración o se ha completado anticipadamente por acuerdo.			

<b>HISTORIA DE USUARIO</b>			
<b>Prioridad:</b> Alta			
<b>CÓDIGO DEL REQUERIMIENTO:</b>	RF12	<b>Actor</b>	Cliente/Entrenador
<b>NOMBRE DEL REQUERIMIENTO</b>	Registro de Avances Semanales.		
<b>Descripción</b>			
Como Cliente/Entrenador, quiero poder registrar avances semanales (peso, grasa corporal, medidas, fotos, comentarios), para monitorear el progreso físico.			
<b>Funcionalidad</b>			
Permitir el registro de métricas físicas y observaciones periódicas para un cliente.			
<b>Criterios de aceptación</b>	1.El sistema debe permitir ingresar peso, grasa corporal, medidas, fotos y comentarios. 2.El sistema debe asociar el registro al cliente y a la fecha. 3.El sistema debe guardar el registro en la base de datos.		
<b>Restricciones</b>			
Los datos deben ser numéricos para peso, grasa y medidas.			

<b>HISTORIA DE USUARIO</b>			
<b>Prioridad: Alta</b>			
<b>CÓDIGO DEL REQUERIMIENTO:</b>	RF13	<b>Actor</b>	Cliente/Entrenador
<b>NOMBRE DEL REQUERIMIENTO</b>	<b>Visualización de Progreso Cronológico</b>		
<b>Descripción</b>			
Como Cliente/Entrenador, quiero poder visualizar mi progreso cronológico, para ver la evolución de mis avances físicos.			
<b>Funcionalidad</b>			
Mostrar un historial de los registros de avances físicos de un cliente en orden cronológico.			
<b>Criterios de aceptación</b>	1. El sistema debe mostrar los registros de avances en un formato fácil de entender (gráficos, tablas). 2. El sistema debe permitir filtrar los registros por rango de fechas. 3.El sistema debe mostrar la fecha de cada registro.		
<b>Restricciones</b>			
Ninguna.			

14.

HISTORIA DE USUARIO			
Prioridad: Baja			
CÓDIGO DEL REQUERIMIENTO:	RF14	Actor	Administrador/Entrenador
NOMBRE DEL REQUERIMIENTO	Eliminación de Registros de Seguimiento Físico.		
Descripción			
Como Administrador/Entrenador, quiero poder eliminar registros de seguimiento físico, para corregir errores o limpiar datos.			
Funcionalidad			
Permitir la eliminación de entradas individuales de seguimiento físico.			
Criterios de aceptación	1. El sistema debe solicitar confirmación antes de eliminar un registro. 2. El sistema debe eliminar el registro seleccionado de la base de datos.		
Restricciones			
La eliminación de un registro puede afectar la visualización del progreso.			

15.

HISTORIA DE USUARIO			
Prioridad: Alta			
CÓDIGO DEL REQUERIMIENTO:	RF15	Actor	Entrenador.
NOMBRE DEL REQUERIMIENTO	Creación de Planes de Alimentación.		
Descripción			
Como Entrenador, quiero poder crear planes de alimentación asociados al cliente y al plan de entrenamiento, para complementar el régimen físico.			
Funcionalidad			
Permitir al entrenador diseñar y asignar planes de alimentación personalizados.			
Criterios de aceptación	1. El sistema debe permitir crear un plan de alimentación con detalles de comidas y porciones. 2. El sistema debe permitir asociar el plan de alimentación a un cliente y a un plan de entrenamiento. 3.El sistema debe guardar el plan de alimentación en la base de datos.		
Restricciones			
Ninguna.			

16.

HISTORIA DE USUARIO			
Prioridad: Alta			
CÓDIGO DEL REQUERIMIENTO:	RNF 16	Actor	Sistema
NOMBRE DEL REQUERIMIENTO	Rollback en Eliminación de Registros de Seguimiento Físico		
Descripción			
Como Sistema, quiero implementar un rollback si la eliminación de registros de seguimiento físico afecta la consistencia del plan, para evitar la pérdida de datos importantes.			
Funcionalidad			
El sistema debe revertir la operación de eliminación si se detecta una inconsistencia.			
Criterios de aceptación	1. Si la eliminación de un registro de seguimiento físico causa una inconsistencia en el plan asociado, la operación debe ser deshecha automáticamente. 2. El sistema debe notificar al usuario si la operación fue revertida.		
Restricciones			
La implementación de rollback debe ser atómica.			

17.

HISTORIA DE USUARIO			
Prioridad: Alta			
CÓDIGO DEL REQUERIMIENTO:	RNF 17	Actor	Sistema.
NOMBRE DEL REQUERIMIENTO	Transacciones para Evitar Inconsistencias en Pagos		
Descripción			
Como Sistema, quiero implementar transacciones para evitar inconsistencias en pagos, para asegurar la integridad de los datos financieros.			
Funcionalidad			
Las operaciones de pago deben ejecutarse como transacciones atómicas.			
Criterios de aceptación	1. Todas las operaciones relacionadas con pagos (registro de ingresos, renovaciones) deben ser transaccionales. 2.Si una parte de la transacción falla, toda la transacción debe ser revertida. 3.El sistema debe garantizar que los balances financieros sean siempre consistentes.		
Restricciones			
Las transacciones deben ser ACID (Atomicidad, Consistencia, Aislamiento, Durabilidad).			

18.

HISTORIA DE USUARIO			
Prioridad: Alta			
CÓDIGO DEL REQUERIMIENTO:	RNF 18	Actor	Sistema.
NOMBRE DEL REQUERIMIENTO	Seguridad de Datos de Clientes.		
Descripción			
Como Sistema, quiero asegurar la seguridad de los datos personales de los clientes, para proteger su privacidad.			
Funcionalidad			
Implementar medidas de seguridad para proteger la información sensible de los clientes.			
Criterios de aceptación	1.Los datos personales de los clientes deben estar encriptados en la base de datos. 2.El acceso a los datos de los clientes debe estar restringido por roles de usuario. 3.El sistema debe cumplir con las normativas de protección de datos.		
Restricciones			
El acceso no autorizado a los datos debe ser imposible.			

19.

HISTORIA DE USUARIO			
Prioridad: Alta			
CÓDIGO DEL REQUERIMIENTO:	RNF 19	Actor	Sistema.
NOMBRE DEL REQUERIMIENTO	Rendimiento de la Aplicación.		
Descripción			
Como Sistema, quiero que la aplicación tenga un rendimiento óptimo, para una experiencia de usuario fluida.			
Funcionalidad			
La aplicación debe responder rápidamente a las interacciones del usuario.			
Criterios de aceptación	1. El tiempo de carga de las páginas no debe exceder los 3 segundos. 2.Las operaciones de búsqueda y listado deben completarse en menos de 2 segundos. 3.La aplicación debe soportar al menos 100 usuarios concurrentes sin degradación del rendimiento.		
Restricciones			
El rendimiento no debe degradarse con el aumento de usuarios o datos.			

20.

HISTORIA DE USUARIO			
Prioridad: Media			
CÓDIGO DEL REQUERIMIENTO:	RNF 20	Actor	Sistema.
NOMBRE DEL REQUERIMIENTO	Disponibilidad del Servicio.		
Descripción			
Como Sistema, quiero que la aplicación esté disponible el 99.9% del tiempo, para que los usuarios puedan acceder a ella cuando la necesiten.			
Funcionalidad			
El servicio debe estar operativo y accesible la mayor parte del tiempo.			
Criterios de aceptación	1. El tiempo de inactividad anual no debe exceder las 8 horas y 45 minutos. 2. El sistema debe tener mecanismos de recuperación ante fallos.		
Restricciones			
El mantenimiento programado debe ser comunicado con antelación.			

## ❖ METODOLOGÍA

Para el cumplimiento del presente proyecto se implementó la metodología Kanban, la cual se centra en la gestión visual del trabajo y en la mejora continua del flujo de actividades. Esta metodología fue seleccionada por su flexibilidad y su enfoque en la entrega constante de valor, ajustándose a las necesidades del equipo y del cliente Gimnasio Los Sayayines.

En la práctica, se utilizó un tablero Kanban en ClickUp dividido en columnas (Por hacer, En progreso, En revisión y Finalizado). Allí se documentaron las historias de usuario, requerimientos funcionales y no funcionales, junto con responsables y tiempos estimados. Este tablero permitió que cada miembro del equipo tuviera claridad sobre el estado de las tareas y facilitó la identificación de cuellos de botella.

### Principios aplicados en el proyecto:

1. **Visualización del trabajo:** todas las tareas se organizaron en el tablero, permitiendo una visión global y transparente del avance del proyecto.
2. **Límites de trabajo en progreso:** se establecieron límites en las columnas “En progreso” y “En revisión” para evitar sobrecarga de tareas y asegurar un flujo constante.

3. **Flujo continuo de valor:** las funcionalidades se desarrollaron y entregaron de manera progresiva, lo que permitió validar avances con el cliente antes de continuar con nuevas implementaciones.
4. **Mejora continua:** a través de reuniones periódicas (daily meetings y retrospectivas), el equipo evaluó su desempeño y ajustó la forma de trabajo para optimizar resultados.

#### **Roles dentro del equipo:**

- **Product Owner:** responsable de priorizar los requerimientos y representar las necesidades del cliente.
- **Team Developers:** encargados del desarrollo técnico de la solución en Node.js y MongoDB, implementando las historias de usuario.
- **Scrum Master:** apoyó en la organización del flujo de trabajo, el seguimiento del tablero Kanban y la eliminación de impedimentos.

#### **Resultados obtenidos:**

El uso de Kanban permitió al equipo mantener una gestión clara y ordenada de las tareas, dar prioridad a los requerimientos más críticos (como la gestión de clientes, planes y finanzas) y garantizar un proceso de desarrollo más ágil, visual y adaptable a cambios. Además, la transparencia en el tablero y las entregas progresivas fortalecieron la comunicación interna y la validación con el cliente.

#### **❖ 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.**

<https://github.com/Sebas404040/SAYAYINES>

- **Link de los videos que se grabaron en las reuniones realizadas (Sprint planning, Daily Stand Up, Sprint Retrospective)**

<https://drive.google.com/file/d/1InnctKxzslDsAE4xrA4ZTHIFGn5oXkl4/view?usp=sharing>





- 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.

BACKLOG

▼ BACKLOG ... + Agregar Tarea

Nombre	Persona asignada	Fecha limite	Prioridad	
▶ ① Crear presentación 3	JG		Baja	...
▶ ① Crear repositorio NPM 3 desarrollo	BV JG		Urgente	...
▶ ① Librerías NPM 1 planeación	BV S JG		Urgente	...
▶ ① Definir Base de datos 2 base de datos	JG		Alta	...
▶ ① Configurar Base de datos 2 base de datos	JG		Urgente	...
▶ ① Desarrollar funcionalidades: CLIENTE 4 desarrollo	BV JG		Urgente	...
▶ ① Desarrollar funcionalidades: PLANES 4 desarrollo	BV JG		Alta	...
▶ ① Desarrollar funcionalidades primarias 2	BV JG		Urgente	...
▶ ① Desarrollar funcionalidades: Contratos 7	BV		Alta	...
▶ ① Desarrollar funcionalidades: Finanzas	JG		Normal	...
▶ ① Definir estructuras 4 planeación	BV JG		Alta	...
▶ ① Desarrollar funcionalidades: Nutrición 5 desarrollo	JG		Alta	...
▶ ① Desarrollar funcionalidades: Seguimiento físico 3 desarrollo	BV JG		Alta	...
▶ ① Documentación en software documentación	S		Normal	...
▶ ① Crear video presentación 2	BV S JG		Baja	...

+ Agregar Tarea



**Link tablero de tareas:**

<https://sharing.clickup.com/90132518580/b/h/6-901320370250-2/d40fbee46e69a68>

**Link BACKLOG:**

<https://sharing.clickup.com/90132518580/b/h/2ky525nm-973/793b627b3918059>

En el proyecto se utilizó un Backlog visual basado en la metodología Kanban, donde se documentaron todas las tareas necesarias para cumplir con los requerimientos funcionales y no funcionales del sistema.

El Backlog incluyó tanto actividades técnicas como administrativas, y permitió organizar el trabajo según prioridad (Urgente, Alta, Normal, Baja), responsables y categorías de tarea.

Principales tareas registradas en el Backlog:

- Planeación y configuración:
  - Definir base de datos
  - Configurar base de datos
  - Definir estructuras
  - Librerías NPM
- Desarrollo de funcionalidades principales:
  - Cliente
  - Planes
  - Contratos



- Finanzas
- Nutrición
- Seguimiento físico
- Soporte y documentación:
  - Documentación en software
  - Crear presentación
  - Crear video presentación
  - Crear repositorio NPM

#### Gestión del Backlog:

- Cada tarea fue asignada a uno o más integrantes del equipo.
- Se definió la prioridad de ejecución para asegurar que los módulos críticos (Cliente, Planes, Contratos y Finanzas) fueran desarrollados primero.
- Se emplearon etiquetas para clasificar el tipo de trabajo (desarrollo, documentación, planeación, base de datos).

De esta manera, el Backlog funcionó como una hoja de ruta clara y visual que permitió cumplir con las metas del proyecto de manera organizada y colaborativa.

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

### **Documentación de Resultados y Funcionamiento del Producto Final**

El producto final desarrollado para el Gimnasio Los Sayayines corresponde a una aplicación web con backend en Node.js y base de datos MongoDB, diseñada para dar solución integral a los problemas identificados en la gestión de clientes, membresías, contratos, seguimiento físico y control financiero.

El sistema implementado permite:



- **Gestión de Clientes:** registro, consulta, actualización y eliminación de clientes, con validación de datos únicos (correo electrónico) y asociación a planes de entrenamiento.
- **Planes de Entrenamiento y Membresía:** creación y administración de planes con duración, metas y niveles; además de la posibilidad de renovarlos, cancelarlos o finalizarlos.
- **Contratos:** generación automática de contratos digitales al asignar un plan, con almacenamiento y seguimiento de condiciones, fechas y estado del contrato.
- **Seguimiento Físico:** registro de avances semanales (peso, grasa corporal, medidas, fotos, comentarios) y visualización del progreso cronológico mediante reportes.
- **Nutrición:** creación de planes de alimentación personalizados asociados al cliente y al plan de entrenamiento.
- **Gestión Financiera:** registro de ingresos y egresos, reportes de balance y transacciones seguras para garantizar la consistencia de los pagos.

#### Tecnologías utilizadas:

- **Node.js:** para la construcción de la API REST que gestiona la lógica del negocio y expone los servicios al frontend.
- **MongoDB:** como base de datos NoSQL para el almacenamiento de información de clientes, planes, contratos y registros de seguimiento físico.
- **Mongoose:** como ODM para la definición de esquemas y validaciones de datos.
- **Git y GitHub:** para el control de versiones y trabajo colaborativo mediante ramas y conventional commits.
- **Kanban (ClickUp):** como herramienta de gestión visual para organizar y dar seguimiento a las tareas del proyecto.

#### Cumplimiento de las tareas planteadas:

El desarrollo se alineó directamente con las **historias de usuario** y los **requerimientos funcionales y no funcionales** definidos en la fase inicial. Cada funcionalidad fue implementada de acuerdo con los criterios de aceptación, asegurando la trazabilidad entre lo solicitado y lo entregado.

- Se completaron las tareas de **gestión de clientes, planes, contratos y finanzas**, cumpliendo con la prioridad establecida en el backlog.
- Los **requerimientos no funcionales**, como seguridad de datos, transacciones financieras y rollback en eliminaciones, fueron documentados y considerados en el diseño de la



aplicación.

- El uso de la metodología **Kanban** facilitó la entrega continua y validación progresiva con el cliente, garantizando que el producto final respondiera a las necesidades planteadas en la situación problema.

### **Resultados:**

El aplicativo desarrollado brinda al Gimnasio Los Sayayines una plataforma centralizada, segura y escalable que mejora la gestión operativa, optimiza los procesos internos y fortalece la experiencia de los clientes. Con ello se cumplió el objetivo principal del proyecto: **resolver los desafíos de gestión a través de una solución tecnológica robusta y eficiente.**

## **7. CONCLUSIONES**

### **Conclusiones generales del documento**

El desarrollo del proyecto para el Gimnasio Los Sayayines permitió demostrar la importancia de aplicar una metodología ágil de gestión como Kanban, ya que facilitó la organización de tareas, la priorización de requerimientos y la entrega progresiva de valor. El uso de Node.js y MongoDB resultó adecuado para construir una solución escalable, segura y eficiente, capaz de centralizar la gestión de clientes, planes, contratos, finanzas y seguimiento físico.

La definición clara de requerimientos funcionales y no funcionales, junto con la redacción de historias de usuario con criterios de aceptación, aseguró que el sistema desarrollado respondiera directamente a las necesidades planteadas en la situación problema. De igual manera, la documentación y uso del backlog visual garantizaron la trazabilidad entre el trabajo planeado y el trabajo entregado.

En conclusión, el proyecto cumplió su objetivo principal: diseñar e implementar una solución tecnológica robusta y adaptable que optimiza los procesos internos del gimnasio y mejora la experiencia de sus clientes.

---

### **Conclusiones de la reunión de retrospectiva del Sprint**

Durante la retrospectiva del sprint, el equipo identificó varios puntos de mejora y aprendizajes importantes:

- **Aspectos positivos:**



- La comunicación constante y el uso de herramientas colaborativas (ClickUp y GitHub) fortalecieron el trabajo en equipo.
- El tablero Kanban permitió visualizar claramente el progreso y evitar sobrecarga en las tareas.
- La validación temprana de funcionalidades con el cliente ayudó a corregir detalles sin generar retrabajo significativo.
- **Aspectos por mejorar:**
  - Ajustar mejor la estimación de tiempos para tareas complejas, ya que algunas demoraron más de lo previsto.
  - Mantener mayor disciplina en las reuniones diarias, evitando desviarse en temas no relacionados con el avance del proyecto.
  - Documentar de forma más detallada los cambios técnicos para facilitar el traspaso de conocimiento entre los miembros del equipo.
- **Compromisos para futuros sprints:**
  - Mejorar la planificación inicial con estimaciones más realistas.
  - Reforzar la automatización de pruebas y respaldos de la base de datos.
  - Continuar aplicando retrospectivas como espacio clave para el aprendizaje y la mejora continua.

En síntesis, la retrospectiva evidenció que el equipo logró un buen desempeño general, pero también resaltó oportunidades de mejora que servirán para proyectos futuros, manteniendo siempre la filosofía de aprendizaje y evolución continua propia de las metodologías ágiles.

