GESTIÓN DE UN GIMNASIO

FABIAN CAMILO PERTUZ TORRES Y CARLOS MARIO VILLAMIZAR MEDINA

SOFTWARE-NODEJS

JUAN CARLOS MARIÑO MORANTES

CAMPUSLANDS
SALON U1
RUTA NODEJS
FLORIDABLANCA
2025
GESTIÓN DE UN GIMNASIO

1. SITUACIÓN PROBLEMA

En la actualidad, muchos entrenadores personales y gimnasios carecen de una herramienta integral que les permita gestionar de forma eficiente a sus clientes y todas las variables relacionadas con su proceso de entrenamiento. Generalmente, la administración de clientes, planes de entrenamiento, seguimientos físicos, nutrición y pagos se lleva a cabo mediante hojas de cálculo, documentos dispersos o incluso registros manuales.

Este enfoque manual y poco centralizado genera múltiples dificultades:

- Pérdida de información relevante sobre el progreso de los clientes.
- Dificultad para asociar rutinas con fechas límite y contratos formales.
- Falta de control financiero claro entre ingresos (mensualidades, sesiones, suplementos) y egresos (gastos operativos, servicios externos).
- Riesgo de inconsistencias al gestionar pagos o cancelaciones, ya que no existen mecanismos de rollback o transacciones que aseguren la integridad de los datos.
- Limitada personalización de la alimentación y seguimiento nutricional, lo cual reduce la efectividad del acompañamiento al cliente.

En consecuencia, se hace necesario desarrollar una aplicación que permita a los entrenadores y gimnasios **centralizar la información de sus clientes, automatizar la gestión de contratos y pagos, registrar avances de forma organizada y garantizar la consistencia de los datos mediante transacciones reales en una base de datos confiable como MongoDB**.

La ausencia de esta herramienta tecnológica provoca ineficiencias administrativas, pérdida de tiempo en tareas manuales, duplicidad de datos y menor calidad en el servicio ofrecido al cliente.

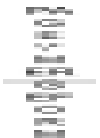
2. LEVANTAMIENTO DE REQUERIMIENTOS

3. REQUERIMIENTOS

3.1. Requerimientos funcionales

La aplicación debe permitir:

Gestión de clientes



- RF1: Crear, listar, actualizar y eliminar clientes.
- RF2: Asociar clientes a uno o varios planes de entrenamiento.

Gestión de planes de entrenamiento

- RF3: Crear planes con nombre, duración, metas físicas y nivel (principiante, intermedio, avanzado).
- RF4: Asociar planes a uno o varios clientes.
- RF5: Registrar un contrato automáticamente al asignar un plan a un cliente.
- RF6: Renovar, cancelar o finalizar un plan de entrenamiento.

Seguimiento físico

- RF7: Registrar avances semanales (peso, grasa corporal, medidas, fotos, comentarios).
- RF8: Consultar el progreso en orden cronológico.
- RF9: Eliminar registros de avances (con rollback si afecta la consistencia del plan).

Nutrición

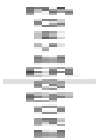
- RF10: Crear planes de alimentación asociados al cliente y al plan de entrenamiento.
- RF11: Registrar alimentos por día con calorías estimadas.
- RF12: Generar reporte nutricional semanal por cliente.

Contratos

- RF13: Generar contrato automático al asignar un plan.
- RF14: El Contrato debe incluir condiciones, duración, precio, fecha inicio y fecha fin.
- RF15: Asociar contrato al cliente y al plan correspondiente.

Gestión financiera

- RF16: Registrar ingresos (mensualidades, sesiones individuales, suplementos).



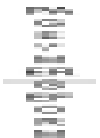
- RF17: Registrar egresos (servicios, gastos operativos).
- RF18: Consultar balance financiero por fecha o cliente.
- RF19: Manejar transacciones reales para evitar inconsistencias en pagos.

3.2. Requerimientos no funcionales

- RNF1: La aplicación debe estar desarrollada completamente en Node.js.
- RNF2: Se debe aplicar Programación Orientada a Objetos.
- RNF3: Cumplir principios Kanban en la organización de clases y módulos.
- RNF4: Implementar al menos dos patrones de diseño (ej. Repository, Factory, Command, Observer).
- RNF5: Persistir datos en MongoDB usando el **driver oficial** (no mongoose).
- RNF6: Implementar operaciones con transacciones reales en MongoDB.
- RNF7: Usar librerías npm para mejorar la experiencia en consola (ej. inquirer, chalk, dotenv, dayjs).
- RNF8: El proyecto debe tener estructura organizada en carpetas (/models, /services, /commands, /config, /utils).
- RNF9: Debe contar con un README.md completo, .gitignore, commits con Conventional Commits.
- RNF10: La planeación debe realizarse con metodología Scrum, documentada en PDF dentro del repositorio.
- RNF11: El repositorio debe ser privado en GitHub e incluir al trainer como colaborador.
- RNF12: Entregar un video de máximo 7 minutos con la explicación técnica y demo funcional.

---

4. HISTORIAS DE USUARIO CON CRITERIOS DE ACEPTACIÓN



1.

HISTORIA DE USUARIO			
Prioridad: Alta			
CÓDIGO DEL REQUERIMIENTO:	RF01	**Actor**	Administrador / Entrenador
NOMBRE DEL REQUERIMIENTO	**Crear, listar, actualizar y eliminar clientes**		
Descripción			
Como administrador quiero gestionar clientes para mantener actualizada la información y poder asociarlos a planes de entrenamiento.			
Funcionalidad			
El sistema debe permitir registrar clientes con datos básicos, consultarlos, modificarlos y eliminarlos (si no tienen planes activos).			
Criterios de aceptación	1. El sistema debe permitir crear un cliente con todos los campos obligatorios. 2. El sistema debe listar clientes existentes. 3. El sistema debe permitir editar datos de un cliente. 4. El sistema debe permitir eliminar un cliente sin planes activos.		
Restricciones			
No se pueden eliminar clientes con planes vigentes.			

2.

HISTORIA DE USUARIO			
Prioridad: Alta			
CÓDIGO DEL REQUERIMIENTO:	RF02	**Actor**	Administrador / Entrenador
NOMBRE DEL REQUERIMIENTO	**Asociar clientes a planes de entrenamiento**		
Descripción			
Como administrador quiero asociar uno o varios planes de entrenamiento a un cliente para gestionar su proceso de forma organizada.			
Funcionalidad			
El sistema debe vincular clientes con planes de entrenamiento activos.			
Criterios de aceptación	1. El sistema debe permitir asignar varios planes a un cliente.		

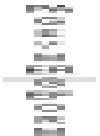
	2. El sistema debe evitar duplicar un mismo plan a un cliente. 3. El sistema debe permitir visualizar los planes activos por cliente.
Restricciones	
Un cliente no puede tener más de un contrato activo para el mismo plan.	

3.

HISTORIA DE USUARIO			
Prioridad: Alta			
CÓDIGO DEL REQUERIMIENTO:	RF03	**Actor**	Administrador / Entrenador
NOMBRE DEL REQUERIMIENTO	**Crear planes de entrenamiento**		
Descripción			
Como administrador quiero crear planes de entrenamiento con nombre, duración, metas y nivel para ofrecer opciones personalizadas a los clientes.			
Funcionalidad			
El sistema debe registrar planes con su información completa.			
Criterios de aceptación	1. El sistema debe permitir registrar un plan con todos los campos obligatorios. 2. El sistema debe validar que el nivel esté entre principiante, intermedio o avanzado. 3. El sistema debe permitir consultar planes existentes.		
Restricciones			
El nombre del plan no debe repetirse.			

4.

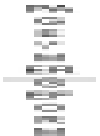
HISTORIA DE USUARIO			
Prioridad: Alta			
CÓDIGO DEL REQUERIMIENTO:	RF04	**Actor**	Administrador
NOMBRE DEL	**Asociar planes a clientes**		



REQUERIMIENTO	
Descripción	
Como administrador quiero asociar planes a clientes para formalizar su proceso de entrenamiento.	
Funcionalidad	
El sistema debe permitir asignar un plan previamente creado a un cliente registrado.	
Criterios de aceptación	1. El sistema debe permitir asignar un plan existente a un cliente. 2. El sistema debe impedir asignar un plan a un cliente inexistente. 3. El sistema debe mostrar el listado de clientes asociados al plan.
Restricciones	
Un plan solo puede asignarse a clientes registrados.	

5.

HISTORIA DE USUARIO			
Prioridad: Alta			
CÓDIGO DEL REQUERIMIENTO:	RF05	**Actor**	Administrador
NOMBRE DEL REQUERIMIENTO	**Contrato automático al asignar plan**		
Descripción			
Como administrador quiero que al asignar un plan a un cliente se genere automáticamente un contrato con condiciones, precio y duración.			
Funcionalidad			
El sistema debe crear un contrato y asociarlo al cliente y al plan.			
Criterios de aceptación	1. El sistema debe generar un contrato al asignar un plan. 2. El contrato debe incluir condiciones, duración, precio y fechas. 3. El contrato debe asociarse al cliente y al plan.		
Restricciones			
Un cliente no puede tener más de un contrato activo para el mismo plan.			

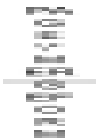


6.

HISTORIA DE USUARIO			
Prioridad: Media			
CÓDIGO DEL REQUERIMIENTO:	RF06	**Actor**	Administrador
NOMBRE DEL REQUERIMIENTO	**Renovar, cancelar o finalizar plan**		
Descripción			
Como administrador quiero poder renovar, cancelar o finalizar un plan para gestionar su ciclo de vida.			
Funcionalidad			
El sistema debe permitir gestionar el estado de un plan asignado.			
Criterios de aceptación	1. El sistema debe permitir renovar un plan vencido. 2. El sistema debe permitir cancelar un plan antes de su vencimiento. 3. El sistema debe permitir finalizar un plan activo.		
Restricciones			
La cancelación debe ejecutar rollback del seguimiento y contrato.			

7.

HISTORIA DE USUARIO			
Prioridad: Alta			
CÓDIGO DEL REQUERIMIENTO:	RF07	**Actor**	Cliente
NOMBRE DEL REQUERIMIENTO	**Registrar avances físicos**		
Descripción			
Como cliente quiero registrar mis avances semanales para medir mi progreso.			
Funcionalidad			
El sistema debe permitir ingresar peso, medidas, grasa corporal, fotos y comentarios.			
Criterios de aceptación	1. El sistema debe permitir ingresar un registro semanal. 2. El sistema debe listar los registros en orden cronológico. 3. El sistema debe permitir eliminar un registro.		



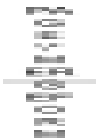
Restricciones
El cliente solo puede registrar avances si tiene un plan activo.

8.

HISTORIA DE USUARIO			
Prioridad: Media			
CÓDIGO DEL REQUERIMIENTO:	RF08	**Actor**	Cliente
NOMBRE DEL REQUERIMIENTO	**Visualizar progreso cronológico**		
Descripción			
Como cliente quiero ver mi progreso en orden cronológico para evaluar mi evolución.			
Funcionalidad			
El sistema debe mostrar los avances en línea de tiempo.			
Criterios de aceptación	1. El sistema debe mostrar los avances desde el primero hasta el último. 2. El sistema debe incluir comentarios y fotos en el historial. 3. El sistema debe mostrar fechas de cada registro.		
Restricciones			
Solo clientes con avances registrados pueden acceder al historial.			

9.

HISTORIA DE USUARIO			
Prioridad: Media			
CÓDIGO DEL REQUERIMIENTO:	RF01	**Actor**	Administrador
NOMBRE DEL REQUERIMIENTO	**Eliminar registros de avances**		
Descripción			
Como administrador quiero eliminar registros de avances cuando sean incorrectos para mantener la consistencia.			
Funcionalidad			



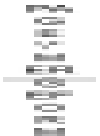
El sistema debe permitir eliminar avances, aplicando rollback si afecta al contrato o plan.	
Criterios de aceptación	1. El sistema debe permitir seleccionar un registro para eliminar. 2. El sistema debe validar si es crítico antes de eliminar. 3. El sistema debe aplicar rollback si es necesario.
Restricciones	
Solo el administrador puede eliminar avances.	

10.

HISTORIA DE USUARIO			
Prioridad: Alta			
CÓDIGO DEL REQUERIMIENTO:	RF10	**Actor**	Administrador
NOMBRE DEL REQUERIMIENTO	**Crear planes de alimentación**		
Descripción			
Como administrador quiero crear planes de alimentación asociados al cliente y al plan de entrenamiento.			
Funcionalidad			
El sistema debe permitir registrar dietas diarias con alimentos y calorías estimadas.			
Criterios de aceptación	1. El sistema debe permitir crear un plan nutricional vinculado a un cliente. 2. El sistema debe asociar el plan a un entrenamiento activo. 3. El sistema debe listar los planes de alimentación creados.		
Restricciones			
Un cliente solo puede tener un plan nutricional si tiene un plan activo.			

11.

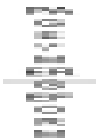
HISTORIA DE USUARIO
Prioridad: Alta



CÓDIGO DEL REQUERIMIENTO:	RF11	**Actor**	Administrador
NOMBRE DEL REQUERIMIENTO	**Generación de reportes financiero**		
Descripción			
Como administrador, puedo generar reportes financieros detallados para conocer los ingresos y egresos del gimnasio en un período de tiempo específico.			
Funcionalidad			
El sistema permitirá seleccionar un rango de fechas y generar un informe que muestre ingresos por mensualidades, planes personalizados, pagos de suplementos y egresos operativos.			
Criterios de aceptación	1. El sistema debe permitir que el administrador seleccione el rango de fechas. 2. El sistema debe mostrar un desglose de ingresos y egresos. 3. El sistema debe permitir exportar el reporte a PDF o Excel.		
Restricciones			
El reporte solo puede generarse si existen registros en el período seleccionado.			

12.

HISTORIA DE USUARIO			
Prioridad: Media			
CÓDIGO DEL REQUERIMIENTO:	RF12	**Actor**	Administrador
NOMBRE DEL REQUERIMIENTO	**Control de stock de suplementos**		
Descripción			
Como administrador, puedo registrar y verificar el pago de mensualidad de cada cliente para mantener actualizado su estado de acceso al gimnasio.			
Funcionalidad			
El sistema permitirá añadir nuevos suplementos, editar su cantidad disponible, registrar ventas y generar alertas cuando el stock esté bajo.			
Criterios de aceptación	1. El sistema debe permitir que el administrador registre suplementos con nombre, precio y stock. 2. El sistema debe disminuir automáticamente el stock tras una venta. 3. El sistema debe generar una alerta si el stock llega al nivel mínimo.		
Restricciones			



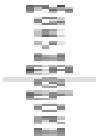
El administrador no podrá registrar un suplemento sin nombre ni precio.

13.

HISTORIA DE USUARIO			
Prioridad: Alta			
CÓDIGO DEL REQUERIMIENTO:	RF13	**Actor**	Administrador
NOMBRE DEL REQUERIMIENTO	**Gestión de pagos de mensualidad**		
Descripción			
Como administrador, puedo registrar y verificar el pago de mensualidad de cada cliente para mantener actualizado su estado de acceso al gimnasio.			
Funcionalidad			
El sistema permitirá registrar pagos de mensualidad indicando fecha, monto y método de pago. Además, actualizará automáticamente el estado del cliente como "activo" o "inactivo".			
Criterios de aceptación	1. El sistema debe permitir registrar pagos con monto, fecha y método. 2. El sistema debe cambiar automáticamente el estado del cliente tras el pago. 3. El sistema debe permitir consultar el historial de pagos de cada cliente. .		
Restricciones			
No se podrá registrar un pago si el cliente no está registrado previamente.			

14.

HISTORIA DE USUARIO			
Prioridad: Alta			
CÓDIGO DEL REQUERIMIENTO:	RF14	**Actor**	Entrenador
NOMBRE DEL REQUERIMIENTO	**Asignación de rutinas personalizadas**		
Descripción			
Como entrenador, puedo asignar una rutina de entrenamiento personalizada a un cliente en función de sus objetivos y características físicas.			
Funcionalidad			



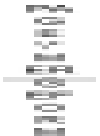
El sistema permitirá que el entrenador seleccione un cliente y le asigne una rutina con ejercicios, repeticiones y días de entrenamiento.	
Criterios de aceptación	1. El sistema debe permitir que el entrenador cree o seleccione rutinas predefinidas. 2. El sistema debe guardar la rutina asociada al cliente. 3. El cliente debe poder visualizar su rutina asignada.
Restricciones	
El entrenador solo puede asignar rutinas a clientes activos.	

15.

HISTORIA DE USUARIO			
Prioridad: Media			
CÓDIGO DEL REQUERIMIENTO:	RF15	**Actor**	Entrenador
NOMBRE DEL REQUERIMIENTO	**Registro de progreso físico del cliente**		
Descripción			
Como entrenador, puedo registrar el progreso físico de un cliente (peso, medidas, porcentaje de grasa, etc.) para evaluar sus avances.			
Funcionalidad			
El sistema permitirá que el entrenador ingrese los datos físicos del cliente en fechas específicas y los almacene en su historial.			
Criterios de aceptación	1. El sistema debe permitir ingresar peso, altura y medidas corporales. 2. El sistema debe guardar los registros con fecha. 3. El sistema debe mostrar gráficas de evolución del cliente.		
Restricciones			
El progreso solo puede ser registrado por un entrenador autorizado.			

16.

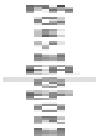
HISTORIA DE USUARIO			
Prioridad: Media			
CÓDIGO DEL REQUERIMIENTO:	RF16	**Actor**	Administrador



NOMBRE DEL REQUERIMIENTO	**Envío de notificaciones de pago**
Descripción	
Como administrador, puedo enviar notificaciones a los clientes cuando su mensualidad esté próxima a vencer o se encuentre vencida.	
Funcionalidad	
El sistema permitirá enviar notificaciones automáticas por correo electrónico o SMS con recordatorios de pago.	
Criterios de aceptación	1. El sistema debe enviar recordatorios 5 días antes de la fecha de vencimiento. 2. El sistema debe enviar un aviso el día del vencimiento. 3. El sistema debe notificar al administrador los clientes morosos.
Restricciones	
El sistema solo enviará notificaciones si el cliente tiene datos de contacto registrados.	

17.

HISTORIA DE USUARIO			
Prioridad: Alta			
CÓDIGO DEL REQUERIMIENTO:	RF17	**Actor**	Administrador
NOMBRE DEL REQUERIMIENTO	**Gestión de roles de usuario**		
Descripción			
Como administrador, puede asignar diferentes roles (administrador, entrenador, cliente) con permisos específicos dentro del sistema.			
Funcionalidad			
El sistema permitirá crear usuarios con roles definidos, controlar el acceso a funcionalidades y garantizar la seguridad de la información.			
Criterios de aceptación	1. El sistema debe permitir asignar un rol a cada usuario. 2. El sistema debe restringir el acceso según el rol. 3. El sistema debe permitir al administrador cambiar roles en cualquier momento.		
Restricciones			



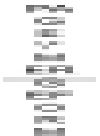
El administrador no puede eliminar su propio rol.

18.

HISTORIA DE USUARIO			
Prioridad: Media			
CÓDIGO DEL REQUERIMIENTO:	RF18	**Actor**	Administrador
NOMBRE DEL REQUERIMIENTO	**Visualización de estadísticas del gimnasio**		
Descripción			
Como administrador, puedo visualizar estadísticas generales sobre clientes, rutinas asignadas, pagos y suplementos vendidos.			
Funcionalidad			
El sistema mostrará gráficas interactivas que representen información como número de clientes activos, rutinas más asignadas, productos vendidos y pagos recibidos.			
Criterios de aceptación	1. El sistema debe mostrar estadísticas en tiempo real. 2. El sistema debe permitir filtrar por períodos. 3. El sistema debe permitir exportar estadísticas en PDF.		
Restricciones			
El administrador debe estar autenticado para acceder a las estadísticas.			

19.

HISTORIA DE USUARIO			
Prioridad: Alta			
CÓDIGO DEL REQUERIMIENTO:	RF19	**Actor**	Administrador
NOMBRE DEL REQUERIMIENTO	**Cancelación de contratos**		
Descripción			
Como administrador, puedo cancelar un contrato de un cliente por solicitud o incumplimiento de pago, con el fin de mantener actualizada la base de datos.			
Funcionalidad			
El sistema permitirá buscar un cliente, cancelar su contrato y dejar registro del motivo de cancelación en el historial.			
Criterios de aceptación	1. El sistema debe permitir al administrador seleccionar el contrato a cancelar.		



	2. El sistema debe solicitar el motivo de cancelación. 3. El sistema debe actualizar el estado del cliente a "inactivo".
Restricciones	
No se podrá cancelar un contrato si el cliente no está registrado en el sistema.	

5. METODOLOGÍA

Para el desarrollo de este proyecto se implementará la metodología **Kanban**, que forma parte de las metodologías ágiles de desarrollo de software. Esta metodología facilita la **visualización del flujo de trabajo**, la **priorización de tareas** y la **entrega continua de valor** al producto.

5.1. Orden de actividades

El flujo de trabajo se gestionará en un tablero Kanban dividido en las siguientes columnas:

1. **Por hacer (To Do):** Contendrá todas las tareas y requerimientos definidos a partir del levantamiento de requerimientos.

2. **En progreso (Doing):** Aquí estarán las tareas en ejecución por parte del equipo de desarrollo.

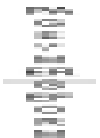
3. **Revisión (Review):** Columna destinada a tareas finalizadas pero pendientes de validación técnica y funcional.

4. **Finalizado (Done):** Una vez revisada y validada, la tarea pasa a esta columna como completada.

Este orden asegura transparencia en el estado del proyecto y permite identificar bloqueos de manera temprana.

5.2. Roles en el equipo

- **Product Owner:**
  Encargado de definir la visión del proyecto, priorizar las funcionalidades en el tablero Kanban y asegurar que el producto cumpla con las necesidades del cliente.



- **Scrum Master:**
  Responsable de facilitar la metodología ágil, asegurar que el equipo trabaje de forma organizada y remover impedimentos que dificulten el avance del proyecto.

- **Developers:**
  Encargados de la implementación técnica del sistema (desarrollo en Node.js, integración con MongoDB, aplicación de principios SOLID y patrones de diseño), pruebas y documentación.

5.3. Definición de Hecho (Definition of Done – DoD)

Una tarea se considera completada únicamente si:

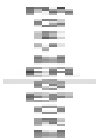
1. El código está implementado y funcional.
2. Cumple los criterios de aceptación definidos en la historia de usuario correspondiente.
3. Ha sido revisado y probado por el equipo de desarrollo.
4. La documentación y commits están actualizados en el repositorio.
5. Se encuentra integrado en la rama principal sin errores.

Con este enfoque, la metodología **Kanban** junto con los roles de **Product Owner, Scrum Master y Developers** garantiza un proceso de desarrollo **organizado, flexible y enfocado en la entrega continua de valor**.

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.
- Link de los videos que se grabaron en las reuniones realizadas (Sprint planning, Daily Stand Up, Sprint Retrospective)
- 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.
- Documentación de los resultados y funcionamiento del producto final resaltando tecnologías utilizadas y cumplimiento de trabajo basado en las tareas planteadas.



7. CONCLUSIONES

7.2. Conclusiones generales.

1. El proyecto plantea una solución integral para la **gestión de clientes, planes de entrenamiento, nutrición, seguimiento físico y finanzas** de un gimnasio, utilizando **Node.js** como tecnología principal.

2. Se estableció una **situación problema clara**, donde se identificó la necesidad de contar con una herramienta que permita digitalizar y organizar la administración de clientes y pagos, garantizando trazabilidad y eficiencia.

3. A través del **levantamiento de requerimientos** y la construcción de **historias de usuario**, se definió un conjunto de funcionalidades que aseguran la cobertura de los procesos esenciales del negocio.

4. La aplicación de **principios SOLID** y **patrones de diseño** garantiza un desarrollo estructurado, mantenible y escalable.

5. La elección de la metodología **Kanban**, junto con roles claramente definidos (Product Owner, Scrum Master y Developers), asegura un proceso ágil, flexible y centrado en la entrega de valor.

7.2. Conclusiones de la retrospectiva del Sprint

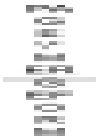
1. El equipo identificó que la organización de las tareas en el tablero Kanban permite **visualizar el flujo de trabajo**, mejorar la comunicación y evitar sobrecargas.

2. Se resaltó la importancia de definir con precisión los **criterios de aceptación** de cada historia de usuario para facilitar las pruebas y validaciones.

3. Durante el sprint, se presentaron algunos bloqueos relacionados con la conexión a la base de datos y la implementación de transacciones, pero se resolvieron gracias al acompañamiento del Scrum Master y la colaboración entre Developers.

4. Se concluyó que la planificación inicial fue adecuada, aunque se debe mejorar la **estimación de tiempos** en tareas críticas como la gestión financiera y la persistencia en MongoDB.

5. El equipo acordó mantener las buenas prácticas de commits con formato **Conventional Commits** y la documentación continua en el repositorio como estándar de calidad.



8. HERRAMIENTA DE SEGUIMIENTO.

La herramienta de seguimiento que se usó, siguiendo la metodología kanban fue Notion, en la cual se distribuyeron las tareas a los miembros del equipo para que según tiempo, prioridad y orden de la tarea se fueran desarrollando progresivamente para así llevar constancia del proyecto y una mejor guia de lo que se había hecho y lo que se tenía que hacer.

Link de los sprints y explicación de la herramienta: https://www.youtube.com/watch?v=PCwlCQC2cZg

---