

Gestor de Portafolio Freelance

**Juan David Santoyo Jaimes
Omar Fernando Granados Parra**

S1

Pedro Felipe Gomez Bonilla

**CAMPUSLANDS
Cajasan
RUTA Node.JS
BUCARAMANGA
2025**

Gestor de Portafolio Freelance

1. SITUACIÓN PROBLEMA

Situación Problema A medida que un profesional freelance crece, la gestión de su negocio se vuelve cada vez más compleja. La información de clientes, proyectos, propuestas y finanzas termina dispersa en múltiples herramientas (hojas de cálculo, correos, gestores de tareas), lo que provoca ineficiencia, pérdida de tiempo y un mayor riesgo de cometer errores.

Tareas cruciales como la creación de contratos, el seguimiento de entregables y el control financiero se convierten en procesos manuales y repetitivos que desvían el enfoque del trabajo creativo y facturable. Esta falta de un sistema centralizado no solo afecta la productividad, sino que también puede dañar la imagen profesional y limitar el potencial de crecimiento.

La necesidad es clara: se requiere una solución unificada que centralice todas las operaciones del negocio freelance, automatice las tareas administrativas y proporcione una visión clara y en tiempo real del estado de cada proyecto y de las finanzas.

2. LEVANTAMIENTO DE REQUERIMIENTOS

1. Requerimientos Funcionales (RF)

Estos requerimientos describen las funcionalidades específicas que el sistema debe ofrecer al usuario.

Módulo de Gestión de Clientes

- RF-01: El sistema debe permitir la creación de nuevos clientes, almacenando como mínimo su nombre, email y teléfono.
- RF-02: El sistema debe permitir la visualización de una lista completa de todos los clientes registrados.
- RF-03: El sistema debe permitir la modificación de los datos de un cliente existente.
- RF-04: El sistema debe permitir la eliminación de un cliente, previa confirmación del usuario.

Módulo de Gestión de Propuestas

- RF-05: El sistema debe permitir la creación de una propuesta asociada a un cliente existente, incluyendo descripción, precio y plazos.
- RF-06: El sistema debe permitir la visualización de todas las propuestas, mostrando su estado (pendiente, aceptada, rechazada).
- RF-07: El sistema debe permitir cambiar el estado de una propuesta. Al cambiar a "aceptada", se debe disparar la creación automática de un proyecto y un contrato.

Módulo de Gestión de Proyectos

- RF-08: El sistema debe permitir la visualización de todos los proyectos, mostrando su estado (activo, pausado, finalizado, cancelado).
- RF-09: El sistema debe permitir ver los detalles completos de un proyecto, incluyendo su contrato asociado y la lista de entregables.
- RF-10: El sistema debe permitir la actualización del estado de un proyecto.

Módulo de Gestión de Entregables

- RF-11: El sistema debe permitir añadir uno o más entregables a un proyecto existente, cada uno con una descripción, fecha límite y estado inicial "pendiente".
- RF-12: El sistema debe permitir la actualización del estado de un entregable (pendiente, entregado, aprobado, rechazado).

Módulo de Gestión Financiera

- RF-13: El sistema debe permitir el registro de ingresos (pagos) asociados a un proyecto.
- RF-14: El sistema debe permitir el registro de egresos (gastos) asociados a un proyecto.
- RF-15: El sistema debe ser capaz de calcular y mostrar un balance financiero (ingresos - egresos) para un proyecto específico.
- RF-16: El sistema debe poder generar un reporte financiero general que totalice todos los ingresos y egresos.

2. Requerimientos No Funcionales (RNF)

Estos requerimientos definen las características de calidad y las restricciones del sistema.

- RNF-01 (Usabilidad): La interfaz de línea de comandos (CLI) debe ser intuitiva, con menús claros y mensajes de feedback que guíen al usuario en todo momento.
- RNF-02 (Rendimiento): Las consultas y operaciones comunes (listar clientes, ver proyectos) deben ejecutarse de forma eficiente, con un tiempo de respuesta inferior a 2 segundos.

- RNF-03 (Fiabilidad): El sistema debe garantizar la integridad de los datos. Las operaciones críticas que involucran múltiples pasos (ej. RF-07) deben ser atómicas, utilizando transacciones de base de datos para evitar estados inconsistentes.
- RNF-04 (Seguridad): La cadena de conexión a la base de datos y otras credenciales sensibles no deben estar escritas en el código fuente. Deben gestionarse a través de variables de entorno (archivo `.env`).
- RNF-05 (Mantenibilidad): El código debe estar estructurado de forma modular y seguir los principios SOLID y los patrones de diseño definidos (Repository, Command) para facilitar su comprensión, modificación y extensión en el futuro.

3. Requerimientos Técnicos

3. La aplicación debe ser desarrollada utilizando Node.js.
4. Se debe aplicar Programación Orientada a Objetos (POO).
5. La interacción con la base de datos se debe realizar con el driver oficial de MongoDB (`mongodb`), sin el uso de ODMs como Mongoose.
6. El proyecto debe ser gestionado con npm y tener un `package.json` bien definido.
7. El código fuente debe estar organizado en un repositorio de Git

4. HISTORIAS DE USUARIO CON CRITERIOS DE ACEPTACIÓN

HISTORIA DE USUARIO			
Prioridad: Alta			
CÓDIGO DEL REQUERIMIENTO:	HU01	Actor	Administrador
NOMBRE DEL REQUERIMIENTO	Creacion de un nuevo cliente		
Descripción			
Como freelancer, quiero poder registrar un nuevo cliente con sus datos principales para tener su información centralizada.			
Funcionalidad			
Permitir al freelancer registrar un nuevo cliente en el sistema. La funcionalidad debe solicitar los datos necesarios, validarlos y persistirlos en la base de datos, confirmando al final la creación exitosa.			
Criterios de aceptación	1. El sistema debe solicitar nombre, email y teléfono del cliente. 2. El campo <code>email</code> debe tener un formato válido y ser único en la base de datos. 3. Si los datos son válidos, el cliente se guarda y se muestra un mensaje de éxito.		
Restricciones			
El administrador solo puede modificar la lista de materiales de una persona si el tipo de persona ya ha sido seleccionado previamente.			

HISTORIA DE USUARIO			
Prioridad: Alta			
CÓDIGO DEL REQUERIMIENTO:	HU02	Actor	Administrador
NOMBRE DEL REQUERIMIENTO	Listado de clientes		
Descripción			
Como freelancer, quiero ver una lista de todos mis clientes para tener una visión general de mi cartera.			
Funcionalidad			
El sistema debe consultar y mostrar en la consola una tabla con la información básica (ID, Nombre, Email) de todos los clientes registrados.			
Criterios de aceptación	1. El sistema debe mostrar una lista de todos los clientes. 2. La lista debe incluir ID, nombre y email. 3. Si no hay clientes, se debe mostrar un mensaje indicándolo.		
Restricciones			
El administrador solo puede modificar la lista de materiales de una persona si el tipo de persona ya ha sido seleccionado previamente.			

HISTORIA DE USUARIO			
Prioridad: Alta			
CÓDIGO DEL REQUERIMIENTO:	HU03	Actor	Administrador
NOMBRE DEL REQUERIMIENTO	Actualización de Datos de Cliente		
Descripción			
Como freelancer, quiero modificar la información de un cliente para mantener sus datos actualizados.			
Funcionalidad			
Permitir al usuario seleccionar un cliente por su ID, modificar sus datos y guardar los cambios en la base de datos.			
Criterios de aceptación	1. El sistema debe permitir buscar un cliente por ID. 2. Si el cliente existe, se deben poder actualizar sus datos. 3. Tras la actualización, el sistema debe mostrar un mensaje de éxito.		
Restricciones			
El administrador solo puede modificar la lista de materiales de una persona si el tipo de persona ya ha sido seleccionado previamente.			

HISTORIA DE USUARIO			
Prioridad: Alta			
CÓDIGO DEL REQUERIMIENTO:	HU04	Actor	Administrador
NOMBRE DEL REQUERIMIENTO	Eliminación de Cliente		
Descripción			
Como freelancer, quiero eliminar un cliente para mantener mi lista de contactos relevante.			
Funcionalidad			
Permitir al usuario seleccionar un cliente por su ID para eliminarlo, previa confirmación para evitar borrados accidentales.			
Criterios de aceptación	1. El sistema debe permitir buscar un cliente por ID. 2. El sistema debe pedir confirmación antes de eliminar. 3. Tras la eliminación, se debe mostrar un mensaje de éxito.		
Restricciones			
El administrador solo puede modificar la lista de materiales de una persona si el tipo de persona ya ha sido seleccionado previamente.			

HISTORIA DE USUARIO			
Prioridad: Alta			
CÓDIGO DEL REQUERIMIENTO:	HU05	Actor	Administrador
NOMBRE DEL REQUERIMIENTO	Creación de Propuesta		
Descripción			
Como freelancer, quiero crear una propuesta para un cliente para formalizar una oferta de servicios.			
Funcionalidad			
Permitir al freelancer crear una propuesta asociándola a un cliente existente, definiendo descripción, precio y plazos.			
Criterios de aceptación	1. El sistema debe permitir seleccionar un cliente existente por ID. 2. Se deben poder ingresar descripción, precio y plazos. 3. La propuesta se debe guardar con estado "pendiente".		
Restricciones			
El administrador solo puede modificar la lista de materiales de una persona si el tipo de persona ya ha sido seleccionado previamente.			

HISTORIA DE USUARIO			
Prioridad: Alta			
CÓDIGO DEL REQUERIMIENTO:	HU06	Actor	Administrador
NOMBRE DEL REQUERIMIENTO	Listado de Propuestas		
Descripción			
Como freelancer, quiero ver todas mis propuestas y su estado para hacer seguimiento comercial.			
Funcionalidad			
El sistema debe mostrar una lista de todas las propuestas, indicando el cliente asociado, el precio y su estado actual (pendiente, aceptada, rechazada).			
Criterios de aceptación	1. El sistema debe mostrar una lista de todas las propuestas. 2. La lista debe incluir ID, nombre del cliente, precio y estado. 3. Si no hay propuestas, se debe mostrar un mensaje		
Restricciones			
El administrador solo puede modificar la lista de materiales de una persona si el tipo de persona ya ha sido seleccionado previamente.			

HISTORIA DE USUARIO			
Prioridad: Alta			
CÓDIGO DEL REQUERIMIENTO:	HU07	Actor	Administrador
NOMBRE DEL REQUERIMIENTO	Aceptación de Propuesta		
Descripción			
Como freelancer, quiero marcar una propuesta como "aceptada" para iniciar un nuevo proyecto formalmente			
Funcionalidad			
Permitir cambiar el estado de una propuesta a "aceptada". Esta acción debe generar automáticamente un nuevo proyecto y un contrato asociado, utilizando una transacción de base de datos.			
Criterios de aceptación	1. El sistema debe permitir seleccionar una propuesta por ID. 2. Al cambiar el estado a "aceptada", se debe crear un nuevo registro en la colección de <code>Proyectos</code> . 3. La operación completa debe ser atómica (o todo se crea o nada se crea).		

Restricciones
El administrador solo puede modificar la lista de materiales de una persona si el tipo de persona ya ha sido seleccionado previamente.

HISTORIA DE USUARIO			
Prioridad: Alta			
CÓDIGO DEL REQUERIMIENTO:	HU08	Actor	Administrador
NOMBRE DEL REQUERIMIENTO	Eliminación de Propuesta		
Descripción			
Como freelancer, quiero eliminar una propuesta para descartar ofertas no aceptadas o incorrectas.			
Funcionalidad			
Permitir al usuario eliminar una propuesta que esté en estado "pendiente" o "rechazada", previa confirmación			
Criterios de aceptación	1. El sistema debe permitir seleccionar una propuesta por ID. 2. Solo se pueden eliminar propuestas en estado "pendiente" o "rechazada". 3. Se debe mostrar un mensaje de éxito tras la eliminación.		
Restricciones			
El administrador solo puede modificar la lista de materiales de una persona si el tipo de persona ya ha sido seleccionado previamente.			

HISTORIA DE USUARIO			
Prioridad: Alta			
CÓDIGO DEL REQUERIMIENTO:	HU09	Actor	Administrador
NOMBRE DEL REQUERIMIENTO	Listado de Proyectos		
Descripción			
Como freelancer, quiero ver todos mis proyectos y su estado para gestionar mi carga de trabajo actual.			
Funcionalidad			
El sistema debe mostrar una lista de todos los proyectos, indicando el cliente asociado y su estado (activo, pausado, finalizado, cancelado).			
Criterios de aceptación	1. El sistema debe mostrar una lista de todos los proyectos. 2. La lista debe incluir ID, nombre del proyecto, cliente y estado. 3. Si no hay proyectos, se debe mostrar un mensaie.		

Restricciones
El administrador solo puede modificar la lista de materiales de una persona si el tipo de persona ya ha sido seleccionado previamente.

HISTORIA DE USUARIO			
Prioridad: Alta			
CÓDIGO DEL REQUERIMIENTO:	HU10	Actor	Administrador
NOMBRE DEL REQUERIMIENTO	Visualización de Detalles de Proyecto		
Descripción			
Como freelancer, quiero ver los detalles de un proyecto específico para consultar toda su información relevante			
Funcionalidad			
Permitir al usuario seleccionar un proyecto por ID y ver toda su información detallada, incluyendo datos del contrato y la lista de entregables asociados.			
Criterios de aceptación	1. El sistema debe permitir seleccionar un proyecto por ID. 2. Se deben mostrar los detalles del proyecto, su contrato y sus entregables. 3. Si el proyecto no se encuentra, se debe mostrar un error.		
Restricciones			
El administrador solo puede modificar la lista de materiales de una persona si el tipo de persona ya ha sido seleccionado previamente.			

HISTORIA DE USUARIO			
Prioridad: Alta			
CÓDIGO DEL REQUERIMIENTO:	HU11	Actor	Administrador
NOMBRE DEL REQUERIMIENTO	Actualización de Estado de Proyecto		
Descripción			
Como freelancer, quiero cambiar el estado de un proyecto para reflejar su progreso (ej. pausar o finalizar).			
Funcionalidad			
Permitir al usuario seleccionar un proyecto por ID y modificar su estado (activo, pausado, finalizado, cancelado).			
Criterios de aceptación	1. El sistema debe permitir seleccionar un proyecto por ID. 2. El usuario debe poder elegir uno de los estados válidos. 3. El cambio debe reflejarse en la base de datos y confirmarse con un mensaje.		
Restricciones			
El administrador solo puede modificar la lista de materiales de una persona si el tipo de persona ya ha sido seleccionado previamente.			

HISTORIA DE USUARIO			
Prioridad: Alta			
CÓDIGO DEL REQUERIMIENTO:	HU12	Actor	Administrador
NOMBRE DEL REQUERIMIENTO	Adición de Entregable a Proyecto		
Descripción			
Como freelancer, quiero añadir entregables a un proyecto para desglosar el trabajo y definir hitos.			
Funcionalidad			
Permitir al usuario seleccionar un proyecto y añadirle un nuevo entregable, especificando una descripción y una fecha límite.			
Criterios de aceptación	1. El sistema debe permitir seleccionar un proyecto por ID. 2. Se debe poder añadir una descripción y fecha límite para el entregable. 3. El entregable se debe guardar con estado "pendiente"		
Restricciones			
El administrador solo puede modificar la lista de materiales de una persona si el tipo de persona ya ha sido seleccionado previamente.			

HISTORIA DE USUARIO			
Prioridad: Alta			
CÓDIGO DEL REQUERIMIENTO:	HU13	Actor	Administrador
NOMBRE DEL REQUERIMIENTO	Listado de Entregables de un Proyecto		
Descripción			
Como freelancer, quiero ver todos los entregables de un proyecto para hacer un seguimiento detallado del avance.			
Funcionalidad			
Al ver los detalles de un proyecto, el sistema debe listar todos sus entregables, mostrando su descripción, fecha límite y estado.			
Criterios de aceptación	1. La vista de detalles del proyecto debe incluir una		

	sección de entregables. 2. Cada entregable en la lista debe mostrar su descripción, fecha límite y estado. 3. Si no hay entregables, se debe indicar.
Restricciones	
El administrador solo puede modificar la lista de materiales de una persona si el tipo de persona ya ha sido seleccionado previamente.	

HISTORIA DE USUARIO			
Prioridad: Alta			
CÓDIGO DEL REQUERIMIENTO:	HU14	Actor	Administrador
NOMBRE DEL REQUERIMIENTO	Actualización de Estado de Entregable		
Descripción			
Como freelancer, quiero marcar un entregable como "entregado" o "aprobado" para registrar el progreso del proyecto.			
Funcionalidad			
Permitir al usuario seleccionar un entregable específico (dentro de un proyecto) y cambiar su estado (pendiente, entregado, aprobado, rechazado).			
Criterios de aceptación	1. El sistema debe permitir seleccionar un entregable por su ID. 2. El usuario debe poder elegir uno de los estados válidos. 3. El cambio debe guardarse y confirmarse con un mensaje.		
Restricciones			
El administrador solo puede modificar la lista de materiales de una persona si el tipo de persona ya ha sido seleccionado previamente.			

HISTORIA DE USUARIO			
Prioridad: Alta			
CÓDIGO DEL REQUERIMIENTO:	HU15	Actor	Administrador
NOMBRE DEL REQUERIMIENTO	Registro de Ingreso		
Descripción			
Como freelancer, quiero registrar un ingreso asociado a un proyecto para llevar un control de los pagos recibidos.			
Funcionalidad			
Permitir al usuario seleccionar un proyecto y registrar un ingreso, especificando el monto y la fecha. La operación debe ser transaccional.			
Criterios de aceptación	1. El sistema debe permitir seleccionar un proyecto por ID.		

	<p>2. Se debe poder ingresar un monto numérico y una fecha.</p> <p>3. El ingreso debe quedar registrado y asociado al proyecto.</p>
Restricciones	
El administrador solo puede modificar la lista de materiales de una persona si el tipo de persona ya ha sido seleccionado previamente.	

HISTORIA DE USUARIO			
Prioridad: Alta			
CÓDIGO DEL REQUERIMIENTO:	HU16	Actor	Administrador
NOMBRE DEL REQUERIMIENTO	Registro de Egreso		
Descripción			
Como freelancer, quiero registrar un gasto asociado a un proyecto para controlar los costos y la rentabilidad.			
Funcionalidad			
Permitir al usuario seleccionar un proyecto y registrar un egreso, especificando el monto, la fecha y una descripción del gasto.			
Criterios de aceptación	1. El sistema debe permitir seleccionar un proyecto por ID. 2. Se debe poder ingresar un monto, fecha y descripción. 3. El egreso debe quedar registrado y asociado al proyecto.		
Restricciones			
El administrador solo puede modificar la lista de materiales de una persona si el tipo de persona ya ha sido seleccionado previamente.			

HISTORIA DE USUARIO			
Prioridad: Alta			
CÓDIGO DEL REQUERIMIENTO:	HU17	Actor	Administrador
NOMBRE DEL REQUERIMIENTO	Actualización de la lista de materiales		
Descripción			
Como freelancer, quiero ver el balance financiero de un proyecto para conocer su rentabilidad.			
Funcionalidad			
Al ver los detalles de un proyecto, el sistema debe calcular y mostrar el total de ingresos, el total de egresos y el balance final (ingresos - egresos).			
Criterios de aceptación	1. La vista de detalles del proyecto debe mostrar el total de ingresos. 2. La vista debe mostrar el total de egresos. 3. La vista debe mostrar el balance final calculado		
Restricciones			
El administrador solo puede modificar la lista de materiales de una persona si el tipo de persona ya ha sido seleccionado previamente.			

HISTORIA DE USUARIO			
Prioridad: Alta			
CÓDIGO DEL REQUERIMIENTO:	HU18	Actor	Administrador
NOMBRE DEL REQUERIMIENTO	Generación de Reporte Financiero		
Descripción			
Como freelancer, quiero generar un reporte de ingresos y egresos para tener una visión general de mis finanzas.			
Funcionalidad			
Permitir al usuario generar un reporte financiero total, sumando todos los ingresos y egresos de todos los proyectos, y mostrando un balance general.			
Criterios de aceptación	1. El sistema debe tener una opción para generar un		

	reporte financiero. 2. El reporte debe mostrar el total de ingresos de todos los proyectos. 3. El reporte debe mostrar el total de egresos de todos los proyectos y el balance general.
Restricciones	
El administrador solo puede modificar la lista de materiales de una persona si el tipo de persona ya ha sido seleccionado previamente.	

5. METODOLOGÍA

Metodología del Proyecto

Este proyecto fue desarrollado siguiendo el marco de trabajo ágil Scrum, lo que permitió un enfoque iterativo, flexible y centrado en la entrega de valor continuo. La metodología se implementó de la siguiente manera:

1. Roles del Equipo Scrum

Para asegurar una correcta ejecución del marco de trabajo, se definieron los siguientes roles:

- **Product Owner: Juan Santoyo**
 - **Responsabilidades:** Definir y priorizar las funcionalidades del producto a través del Product Backlog, asegurar que el equipo de desarrollo entienda los requerimientos y maximizar el valor del trabajo entregado.
- **Scrum Master: Omar Granados**
 - **Responsabilidades:** Facilitar los eventos de Scrum, eliminar impedimentos que bloqueen al equipo y asegurar que el marco de trabajo se entienda y se aplique correctamente.
- **Equipo de Desarrollo: Juan Santoyo, Omar Granados**
 - **Responsabilidades:** Desarrollar el incremento del producto, auto-organizarse para completar el trabajo del Sprint y garantizar la calidad técnica de la entrega.

2. Proceso de Desarrollo y Artefactos

El flujo de trabajo se organizó en Sprints de una semana, siguiendo los eventos y artefactos clave de Scrum:

- **Product Backlog:** Se inició con un levantamiento de requerimientos (funcionales y no funcionales) que sirvió de base para crear un Product Backlog detallado. Este backlog se compuso de 18 Historias de Usuario, cada una con su descripción, funcionalidad y criterios de aceptación, asegurando una visión clara del producto a construir.

- **Sprint Planning:** Al inicio de cada Sprint, el equipo realizaba una reunión de planificación para seleccionar un conjunto de Historias de Usuario del Product Backlog. Estas historias conformaban el Sprint Backlog, que representaba el objetivo a cumplir durante esa iteración.
- **Ejecución del Sprint:** Durante el Sprint, el equipo de desarrollo trabajaba en las tareas definidas para completar las historias de usuario seleccionadas. Se realizaban reuniones diarias (Daily Scrums) para sincronizar el progreso y planificar el trabajo de las siguientes 24 horas.
- **Sprint Review:** Al finalizar cada Sprint, se presentaba el incremento de software funcional a los stakeholders (en este caso, al Product Owner) para obtener feedback y validar que el trabajo cumplía con lo esperado.
- **Sprint Retrospective:** Tras la revisión, el equipo realizaba una retrospectiva para reflexionar sobre el Sprint, identificar qué funcionó bien, qué se podría mejorar y definir acciones concretas para el siguiente ciclo.

3. Herramientas Utilizadas

- **Gestión de Proyecto:** Se utilizó Trello para gestionar el Product Backlog y el Sprint Backlog. Se creó un tablero con columnas (Product Backlog, Sprint Backlog, En Progreso, Hecho) para visualizar el flujo de trabajo y el estado de cada historia de usuario.
- **Control de Versiones:** Se utilizó Git y GitHub para el control de versiones del código fuente, facilitando la colaboración y el seguimiento de cambios.
- **Documentación:** La documentación principal del proyecto, incluyendo la descripción, requerimientos y metodología, se centralizó en el archivo `README.md`

6. EVIDENCIA DE PLANTEAMIENTO DE PLATAFORMA DE TRABAJO

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

- https://github.com/JuanSantoyoJ/Proyecto_S1_Node.JS_SantoyoJuan_GradosOmar
- <https://trello.com/b/Wubqyf4B/gestor-de-portafolio-freelance>
- Documentación de los resultados y funcionamiento del producto final resaltando tecnologías utilizadas y cumplimiento de trabajo basado en las tareas planteadas.

7. CONCLUSIONES

El desarrollo de este Gestor de Portafolio de Proyectos Freelance ha permitido alcanzar con éxito los objetivos planteados, resultando en una herramienta de línea de comandos funcional y robusta que soluciona la

problemática de la gestión desorganizada para los profesionales independientes.

A lo largo del proyecto, se han extraído las siguientes conclusiones clave:

1. **Impacto de la Arquitectura de Software:** La aplicación de los principios SOLID y los patrones de diseño Repository y Command fue fundamental para construir un sistema mantenible, escalable y fácil de probar.
 - El patrón Repository demostró ser altamente efectivo para desacoplar la lógica de negocio del acceso a la base de datos, lo que simplifica futuras modificaciones y facilita las pruebas unitarias.
 - El patrón Command permitió encapsular cada acción del usuario en objetos, resultando en un código más limpio y organizado, donde las responsabilidades están claramente definidas.
2. **Beneficios de la Metodología Scrum:** La adopción de Scrum como marco de trabajo ágil fue crucial para el éxito del proyecto. La planificación en Sprints, la definición clara de roles y el uso de herramientas como Trello permitieron una gestión organizada, una adaptación rápida a los desafíos y una entrega de valor constante.
3. **Tecnologías y Herramientas:** El uso de Node.js se confirmó como una elección acertada para el desarrollo de una aplicación de CLI potente y eficiente. La utilización del driver nativo de MongoDB y la implementación de transacciones fueron esenciales para garantizar la integridad y consistencia de los datos en las operaciones financieras y de gestión más críticas.
4. **Aprendizaje y Crecimiento:** Este proyecto no solo ha sido un ejercicio técnico, sino también una valiosa experiencia en la planificación, gestión y ejecución de un producto de software de principio a fin, siguiendo las mejores prácticas de la industria.

Posibles Mejoras a Futuro

El sistema actual sienta una base sólida que puede ser extendida con nuevas funcionalidades, tales como:

- Generación de reportes en PDF para contratos y facturas.
- Notificaciones por correo electrónico para recordatorios de fechas de entrega o pagos.
- Dashboard de métricas para visualizar la rentabilidad y el rendimiento a lo largo del tiempo.
- Evolución a una interfaz web para ofrecer una experiencia de usuario