

Historias de Usuario

PROYECTO NODE.JS

Andrés David Reyes Espinel

Vladimir Díaz Contreras

Edgar Leonardo Acevedo Arteaga

GRUPO S1 CAJASAN

DOCENTE

INGENIERO PEDRO FELIPE GÓMEZ BONILLA

**CAMPUSLANDS
SALÓN: S1
RUTA: Node.js
BUCARAMANGA
2025**

Node.js

1. SITUACIÓN PROBLEMA

Una empresa de freelancer que maneja distintos proyectos de clientes enfrenta dificultades para organizar de manera eficiente la información, ya que anteriormente la empresa utilizaba hojas de cálculo y documentos dispersos lo cual genera pérdida de información, duplicidad de registros, falta de control sobre el estado de las propuestas y proyectos, y errores en la gestión de ingresos y egresos. Por ello, se requiere una aplicación de línea de comandos (CLI) en **Node.js** que permita centralizar y automatizar la gestión de su portafolio profesional, garantizando un control confiable y organizado de todo el proceso.

2. LEVANTAMIENTO DE REQUERIMIENTOS

El presente documento tiene como objetivo recopilar y definir los requerimientos necesarios para el desarrollo de una aplicación de línea de comandos en Node.js que permita a una empresa de freelancers gestione de manera centralizada y automatizada su portafolio profesional. Esta herramienta busca sustituir el uso disperso de hojas de cálculo y documentos que actualmente dificultan la organización eficiente de la información.

La aplicación cubrirá la gestión integral de clientes, propuestas, proyectos, contratos, entregables y finanzas, permitiendo registrar ingresos y egresos, generar contratos y controlar el progreso de los entregables, con el fin de evitar duplicidad, pérdida de información y mejorar la productividad del equipo freelance.

3. REQUERIMIENTOS

3.1. Requerimientos Funcionales

1. Gestión de clientes
 - Registrar nuevos clientes
 - Consultar clientes registrados
 - Actualizar información de un cliente

- Eliminar clientes
- Asociar clientes a proyectos y propuestas
- 2. Gestión de propuestas
 - Crear propuestas para clientes con descripción, precio, plazos y estado (*pendiente, aceptada, rechazada*).
 - Consultar propuestas por cliente o estado.
 - Actualizar o eliminar propuestas.
 - Generar un proyecto automáticamente cuando una propuesta es aceptada.
- 3. Gestión de proyectos
 - Crear, listar, actualizar y eliminar proyectos.
 - Asociar un proyecto con cliente, contrato y entregables.
 - Cambiar el estado del proyecto (*activo, pausado, finalizado, cancelado*).
 - Registrar avances de proyectos.
- 4. Gestión de contratos
 - Generar un contrato asociado a un proyecto.
 - Registrar condiciones, fechas de inicio/fin y valor total.
 - Consultar contratos asociados a proyectos.
- 5. Gestión de entregables
 - Registrar entregables con fecha límite y estado (*pendiente, entregado, aprobado, rechazado*).
 - Actualizar o eliminar entregables.
 - Controlar los cambios de estado de entregables con posibilidad de rollback mediante transacciones.
- 6. Gestión financiera
 - Registrar ingresos y egresos asociados a proyectos.
 - Consultar balance financiero por cliente o rango de fechas.
 - Usar transacciones para registrar pagos y evitar inconsistencias.
- 7. Interfaz CLI
 - Menús interactivos para acceder a cada módulo (clientes, propuestas, proyectos, contratos, entregables, finanzas). Mensajes de error, advertencia y confirmación claros en consola.

3.2. Requerimientos No Funcionales

1. Usabilidad

- La aplicación debe ofrecer una interfaz de consola intuitiva y fácil de usar, con menús interactivos y mensajes claros.

2. Rendimiento

- La aplicación al ejecutar tareas como consultas, creación y actualización de datos deben ejecutarse en menos de 3 segundos en condiciones.

3. Fiabilidad y consistencia

- El sistema debe garantizar que no existan registros duplicados o con datos incompletos.
- Las operaciones críticas (ej. registro de pagos o actualización de entregables) deben realizarse dentro de transacciones en MongoDB para garantizar la consistencia de los datos.

4. Escalabilidad

- La arquitectura debe permitir agregar nuevas funcionalidades sin modificar drásticamente el código existente, aplicando principios SOLID y patrones de diseño.

5. Confiabilidad

- El sistema debe manejar adecuadamente errores de conexión, validaciones de datos y excepciones durante la ejecución para evitar caídas inesperadas.

4. HISTORIAS DE USUARIO CON CRITERIOS DE ACEPTACIÓN

HISTORIA DE USUARIO			
Prioridad: Alta			
CÓDIGO DEL REQUERIMIENTO:	RF01	Actor	Administrador
NOMBRE DEL REQUERIMIENTO	Registrar nuevos clientes		
Descripción			
Como administrador, puedo registrar nuevos clientes en el sistema ingresando su información básica (nombre, identificación, datos de contacto), con el fin de mantener un registro organizado y disponible para asociarlos posteriormente a propuestas y proyectos			
Funcionalidad			
Permitir al administrador registrar nuevos clientes en el sistema, ingresando datos como nombre, identificación, información de contacto y empresa, asegurando que la información sea validada antes de almacenarse y quedando disponible para su posterior consulta, actualización o asociación con proyectos y propuestas.			
Criterios de aceptación	<ol style="list-style-type: none"> 1. El sistema debe permitir que el administrador ingrese la información básica del cliente (nombre, identificación, datos de contacto y empresa). 2. El sistema debe validar que los campos requeridos no estén vacíos y que los formatos sean correctos (ejemplo: número de identificación único, correo válido). 3. El sistema debe guardar la información del cliente en la base de datos de manera persistente. 4. El sistema debe notificar al administrador con un mensaje de confirmación cuando el cliente se registre correctamente o mostrar un mensaje de error si falla el proceso. 		
Restricciones			
<ul style="list-style-type: none"> • El administrador solo puede registrar un cliente si completa todos los campos obligatorios. • No se permite registrar dos clientes con la misma identificación o correo electrónico. • El administrador no puede modificar los datos de un cliente durante el proceso de registro; los cambios solo podrán hacerse desde la opción de actualización. • El sistema debe rechazar registros con formatos inválidos (ejemplo: correo sin formato válido, identificación no numérica). 			

HISTORIA DE USUARIO			
Prioridad: A			
lta			
CÓDIGO DEL REQUERIMIENTO:	RF02	Actor	Administrador
NOMBRE DEL REQUERIMIENTO	Consultar clientes registrados		
Descripción			
Como administrador, puedo consultar los clientes registrados en el sistema para visualizar su información básica y confirmar que estén disponibles para asociarlos a propuestas o proyectos.			
Funcionalidad			
Permitir al administrador consultar los clientes previamente registrados, mostrando en la consola datos relevantes como nombre, identificación, datos de contacto y empresa, con la posibilidad de recorrer toda la lista de manera ordenada.			
Criterios de aceptación	<ol style="list-style-type: none"> 1. El sistema debe mostrar al administrador todos los clientes registrados en la base de datos. 2. El sistema debe presentar la información de cada cliente en un formato claro y legible. 3. El sistema debe notificar cuando no existan clientes registrados. 4. La información consultada debe coincidir fielmente con la almacenada en la base de datos. 		
Restricciones			
<ul style="list-style-type: none"> • Solo el administrador puede acceder a la lista de clientes registrados. • La consulta únicamente mostrará clientes que ya existan en la base de datos. • Desde la opción de consulta no es posible editar ni eliminar información. 			

HISTORIA DE USUARIO			
Prioridad: Alta			
CÓDIGO DEL REQUERIMIENTO:	RF03	Actor	Administrador
NOMBRE DEL REQUERIMIENTO	Actualización de la lista de materiales		
Descripción			
Como administrador, puedo actualizar la información de un cliente ya registrado, con el fin de corregir datos erróneos o mantener actualizados los datos de contacto y empresa asociados.			
Funcionalidad			
Permitir al administrador seleccionar un cliente registrado y modificar su información (nombre, identificación, contacto o empresa), validando los cambios antes de guardarlos en la base de datos y garantizando que no se generen duplicados ni inconsistencias.			
Criterios de aceptación	<ol style="list-style-type: none"> 1. El sistema debe permitir seleccionar al cliente a actualizar a partir de la lista de clientes registrados. 2. El sistema debe validar que los nuevos datos ingresados cumplan con los formatos y reglas establecidos (ejemplo: correo válido, identificación única). 3. El sistema debe almacenar los cambios en la base de datos de manera persistente. 4. El sistema debe notificar al administrador si la actualización se realizó con éxito o si ocurrió algún error. 		
Restricciones			
<ul style="list-style-type: none"> • Solo el administrador puede actualizar la información de los clientes. • No se permite modificar la identificación de un cliente a un valor ya existente en otro registro. • La actualización no debe permitir dejar campos obligatorios vacíos. 			

HISTORIA DE USUARIO			
Prioridad: Alta			
CÓDIGO DEL REQUERIMIENTO:	RF04	Actor	Administrador
NOMBRE DEL REQUERIMIENTO	Eliminar clientes		
Descripción			
Como administrador, puedo eliminar clientes del sistema para mantener la base de datos limpia y evitar registros innecesarios o duplicados.			
Funcionalidad			
Permitir al administrador seleccionar un cliente previamente registrado y eliminarlo de la base de datos, garantizando que la acción se confirme antes de ejecutarse para evitar borrados accidentales.			
Criterios de aceptación	<ol style="list-style-type: none"> 1. El sistema debe mostrar la lista de clientes registrados para que el administrador seleccione cuál eliminar. 2. El sistema debe solicitar confirmación antes de proceder con la eliminación. 3. El sistema debe eliminar el cliente seleccionado de la base de datos de manera persistente. 4. El sistema debe notificar al administrador si la eliminación fue exitosa o si ocurrió un error. 		
Restricciones			
<ul style="list-style-type: none"> • Solo el administrador puede eliminar clientes. • No se debe permitir la eliminación de un cliente que tenga proyectos o contratos activos asociados. • Una vez eliminado, el cliente no podrá recuperarse (a menos que se implemente un historial o backup). 			

HISTORIA DE USUARIO			
Prioridad: Alta			
CÓDIGO DEL REQUERIMIENTO:	RF05	Actor	Administrador
NOMBRE DEL REQUERIMIENTO	Asociar clientes a proyectos y propuestas		
Descripción			
Como administrador, puedo asociar clientes registrados a proyectos y propuestas, con el fin de mantener un vínculo claro entre la información del cliente y el trabajo que se le desarrolla.			
Funcionalidad			
Permitir al administrador seleccionar un cliente registrado y vincularlo a una propuesta o proyecto, de manera que toda la información quede relacionada en la base de datos y sea posible rastrear fácilmente qué cliente corresponde a cada iniciativa.			
Criterios de aceptación	<ol style="list-style-type: none"> 1. El sistema debe permitir seleccionar un cliente existente al momento de crear una propuesta o un proyecto. 2. El sistema debe validar que el cliente esté registrado previamente antes de asociarlo. 3. El sistema debe guardar la relación cliente–propuesta o cliente–proyecto en la base de datos. 4. El sistema debe notificar al administrador cuando la asociación se realice con éxito o mostrar un error en caso de fallar. 		
Restricciones			
<ul style="list-style-type: none"> • Solo el administrador puede asociar clientes a propuestas o proyectos. • No se puede asociar un cliente inexistente o eliminado. • Cada propuesta o proyecto debe estar obligatoriamente vinculado a un cliente válido. 			

HISTORIA DE USUARIO			
Prioridad: Alta			
CÓDIGO DEL REQUERIMIENTO:	RF06	Actor	Administrador
NOMBRE DEL REQUERIMIENTO	Crear propuestas para clientes con descripción, precio, plazos y estado (<i>pendiente, aceptada, rechazada</i>).		
Descripción			
Como administrador, puedo crear propuestas para clientes que incluyan una descripción del trabajo, precio, plazos de entrega y estado (pendiente, aceptada o rechazada), con el fin de formalizar y organizar la oferta de servicios al cliente.			
Funcionalidad			
Permitir al administrador registrar una nueva propuesta vinculada a un cliente existente, ingresando detalles como la descripción del proyecto, el costo estimado, los plazos de entrega y el estado inicial de la propuesta. Estos datos deben almacenarse en la base de datos para su posterior consulta, actualización o seguimiento.			
Criterios de aceptación	<ol style="list-style-type: none"> 1. El sistema debe permitir seleccionar un cliente existente al que se le asignará la propuesta. 2. El sistema debe solicitar los campos obligatorios: descripción, precio, plazos y estado. 3. El sistema debe validar los formatos (precio numérico, plazos en fechas válidas, estado dentro de los valores permitidos). 4. El sistema debe guardar la propuesta en la base de datos de manera persistente. 5. El sistema debe confirmar al administrador que la propuesta fue creada exitosamente o notificar en caso de error. 		
Restricciones			
<ul style="list-style-type: none"> • Solo el administrador puede crear propuestas. • No se pueden registrar propuestas sin un cliente asociado. • El estado inicial por defecto debe ser pendiente, aunque puede ser modificado a aceptada o rechazada posteriormente. • No se permite ingresar valores negativos en el precio ni plazos inválidos. 			

HISTORIA DE USUARIO			
Prioridad: Alta			
CÓDIGO DEL REQUERIMIENTO:	RF07	Actor	Administrador
NOMBRE DEL REQUERIMIENTO	Consultar propuestas por cliente o estado.		
Descripción			
Como administrador, quiero consultar las propuestas registradas filtrando por cliente o por estado (pendiente, aceptada, rechazada), con el fin de realizar seguimiento, control y toma de decisiones sobre las ofertas de servicios.			
Funcionalidad			
Permitir al administrador buscar y visualizar las propuestas almacenadas en la base de datos, aplicando filtros por cliente y/o estado. El sistema debe mostrar en la consulta los datos relevantes de cada propuesta (cliente, descripción, precio, plazos y estado actual). Además, debe permitir combinar filtros y mostrar los resultados de forma clara y organizada.			
Criterios de aceptación		<ol style="list-style-type: none"> 1. El sistema debe permitir filtrar las propuestas por cliente seleccionado. 2. El sistema debe permitir filtrar las propuestas por estado (pendiente, aceptada, rechazada). 3. El sistema debe permitir aplicar filtros combinados (cliente + estado). 4. El sistema debe mostrar una lista con la información básica de las propuestas (cliente, descripción, precio, plazos y estado). 5. El sistema debe notificar al administrador si no se encuentran propuestas con los filtros aplicados. 6. El sistema debe recuperar la información desde la base de datos de manera persistente. 	
Restricciones			
<ul style="list-style-type: none"> • Solo el administrador puede consultar propuestas. • No se debe permitir consultar propuestas si no existen clientes registrados. • El sistema solo debe mostrar propuestas con estados válidos (pendiente, aceptada, rechazada). • Los filtros de búsqueda deben ser obligatorios (al menos uno: cliente o estado). 			

HISTORIA DE USUARIO			
Prioridad: Alta			
CÓDIGO DEL REQUERIMIENTO:	RF08	Actor	Administrador
NOMBRE DEL REQUERIMIENTO	Actualizar o eliminar propuestas.		
Descripción			
Como administrador, quiero actualizar o eliminar propuestas previamente registradas para corregir información o descartar aquellas que ya no sean necesarias, con el fin de mantener la base de datos consistente y actualizada.			
Funcionalidad			
El sistema debe permitir al administrador actualizar los datos de una propuesta existente, lo que incluye la posibilidad de modificar la descripción, el precio, los plazos y el estado (pendiente, aceptada o rechazada). Asimismo, el administrador debe tener la opción de eliminar propuestas previamente registradas de forma permanente en la base de datos. En ambos casos, el sistema debe validar que los cambios realizados cumplan con los formatos y reglas establecidas, tales como que el precio sea numérico y positivo, que los plazos correspondan a fechas válidas y que el estado pertenezca únicamente a los valores permitidos. Finalmente, el sistema debe confirmar al administrador cuando la propuesta se haya actualizado o eliminado de manera exitosa, o en caso contrario, notificar los errores encontrados durante la operación.			
Criterios de aceptación	<ol style="list-style-type: none"> 1. El sistema debe permitir seleccionar una propuesta existente para actualizar o eliminar. 2. El sistema debe validar los campos al actualizar (precio positivo, plazos válidos, estado permitido). 3. El sistema debe registrar en la base de datos los cambios aplicados a la propuesta. 4. El sistema debe eliminar de manera persistente la propuesta seleccionada en caso de confirmación. 5. El sistema debe notificar al administrador cuando una propuesta se haya actualizado o eliminado exitosamente. 6. El sistema debe mostrar un mensaje de error si la propuesta no existe o si los datos ingresados son inválidos. 		
Restricciones			
<ul style="list-style-type: none"> • Solo el administrador puede actualizar o eliminar propuestas. • No se permite actualizar una propuesta sin cliente asociado. • No se permite ingresar valores negativos en precio ni plazos inválidos. • Una propuesta eliminada no puede recuperarse posteriormente. • El estado de la propuesta solo puede ser pendiente, aceptada o rechazada. 			

HISTORIA DE USUARIO			
Prioridad: Alta			
CÓDIGO DEL REQUERIMIENTO:	RF09	Actor	Administrador
NOMBRE DEL REQUERIMIENTO	Generar un proyecto automáticamente cuando una propuesta es aceptada		
Descripción			
Como administrador, quiero que el sistema genere automáticamente un proyecto cuando una propuesta sea aceptada, de modo que se cree un registro formal del proyecto asociado al cliente y la propuesta, evitando duplicidad de registros y asegurando la continuidad en la gestión.			
Funcionalidad			
El sistema debe crear automáticamente un proyecto en el módulo de proyectos una vez que una propuesta cambie de estado a "Aceptada" , vinculando al cliente correspondiente y heredando información básica de la propuesta (descripción, precio, plazos y estado inicial del proyecto). De esta manera, se garantiza que toda propuesta <u>aceptada</u> tenga un proyecto formalmente registrado para su gestión posterior.			
Criterios de aceptación	<ol style="list-style-type: none"> 1. El sistema debe detectar cuando una propuesta cambia su estado a "Aceptada". 2. El sistema debe crear automáticamente un proyecto con la información básica heredada de la propuesta (cliente, descripción, precio, plazos). 3. El sistema debe registrar al proyecto con un estado inicial definido (ejemplo: "En planificación"). 4. El sistema debe generar un identificador único para el nuevo proyecto y almacenarlo en la base de datos. 5. El sistema debe notificar al administrador que el proyecto ha sido creado correctamente o mostrar un mensaje de error si el proceso falla. 		
Restricciones			
<ul style="list-style-type: none"> • No se debe permitir la creación manual de proyectos a partir de propuestas aceptadas (solo de forma automática). • Si una propuesta ya generó un proyecto previamente, el sistema no debe duplicar el registro. • La creación del proyecto debe heredar solo información válida de la propuesta; si los datos están incompletos o con formatos incorrectos, el sistema debe rechazar la acción. • El estado inicial del proyecto debe estar predefinido y no editable durante la creación automática. 			

HISTORIA DE USUARIO			
Prioridad: Alta			
CÓDIGO DEL REQUERIMIENTO:	RF10	Actor	Administrador
NOMBRE DEL REQUERIMIENTO	Crear, listar, actualizar y eliminar proyectos		
Descripción			
Como administrador, quiero que el sistema genere automáticamente un proyecto cuando una propuesta sea aceptada, de modo que se cree un registro formal del proyecto asociado al cliente y la propuesta, evitando duplicidad de registros y asegurando la continuidad en la gestión.			
Funcionalidad			
El sistema debe permitir al administrador gestionar de manera completa los proyectos registrados. Para ello, debe ofrecer la posibilidad de crear nuevos proyectos ingresando la información básica como el nombre, el cliente asociado, la propuesta relacionada, las fechas de inicio y finalización, el presupuesto y el estado inicial. Así mismo, debe mostrar un listado con todos los proyectos existentes, facilitando su consulta mediante filtros por estado, cliente o fecha de creación. También debe ser posible actualizar la información de los proyectos ya registrados, permitiendo modificar datos como las fechas, la descripción, el presupuesto o el estado, asegurando que siempre se mantenga información precisa y actualizada. Finalmente, el sistema debe brindar la opción de eliminar proyectos que no se encuentren activos o en ejecución, con el fin de evitar duplicidad o registros innecesarios, notificando al administrador cuando la operación se realice con éxito o en caso de presentarse algún error.			
Criterios de aceptación	<ol style="list-style-type: none"> 1. El sistema debe permitir crear proyectos registrando información básica obligatoria (nombre, cliente, propuesta, fechas y estado). 2. El sistema debe validar que los campos obligatorios no estén vacíos y que los formatos sean correctos (ejemplo: fechas válidas, valores numéricos positivos). 3. El sistema debe mostrar un listado de proyectos, permitiendo filtrar por cliente, estado o fecha. 4. El sistema debe permitir actualizar la información de un proyecto y guardar los cambios correctamente en la base de datos. 5. El sistema debe permitir eliminar proyectos y notificar al administrador si el proceso fue exitoso o si no se puede realizar (ejemplo: proyectos activos en ejecución). 6. El sistema debe emitir mensajes claros de confirmación o error en cada operación (crear, listar, actualizar, eliminar). 		
Restricciones			
<ul style="list-style-type: none"> • El administrador solo puede crear proyectos si la propuesta asociada ya está registrada y aceptada. • No se deben eliminar proyectos en ejecución o finalizados, solo los que no hayan iniciado. • Los proyectos eliminados deben marcarse como inactivos en la base de datos para mantener trazabilidad (no borrarse físicamente). • Los datos de presupuesto deben ser numéricos y positivos. • El sistema debe impedir registros duplicados (mismo cliente, mismo nombre y fechas idénticas). 			

HISTORIA DE USUARIO			
Prioridad: Alta			
CÓDIGO DEL REQUERIMIENTO:	RF11	Actor	Administrador
NOMBRE DEL REQUERIMIENTO	Asociar un proyecto con cliente, contrato y entregables		
Descripción			
Como administrador, quiero que el sistema permita asociar un proyecto con el cliente correspondiente, el contrato firmado y los entregables definidos, con el fin de mantener la trazabilidad y asegurar que cada proyecto tenga vinculada toda la información necesaria para su gestión y seguimiento.			
Funcionalidad			
El sistema debe permitir al administrador relacionar cada proyecto con el cliente al que pertenece, vinculando automáticamente los datos del cliente que ya se encuentran registrados en la base de datos. Asimismo, debe permitir asociar el contrato que formaliza el acuerdo entre el cliente y la empresa, asegurando que el documento quede registrado y ligado al proyecto. Adicionalmente, el sistema debe ofrecer la posibilidad de registrar y asociar los entregables acordados, indicando su descripción, plazos de entrega y estado de cumplimiento. Esta funcionalidad debe garantizar que la información se mantenga organizada y disponible para su consulta, evitando inconsistencias y asegurando que cada proyecto cuente con la documentación y compromisos formales correspondientes.			
Criterios de aceptación	<ol style="list-style-type: none"> 1. El sistema debe permitir asociar un proyecto con un cliente existente en la base de datos. 2. El sistema debe validar que cada proyecto tenga un contrato registrado antes de marcarlo como activo. 3. El sistema debe permitir registrar y asociar entregables al proyecto, incluyendo descripción, fechas y estado. 4. El sistema debe guardar correctamente las asociaciones en la base de datos y mantener la integridad de la información. 5. El sistema debe mostrar mensajes de confirmación al realizar la asociación y notificar al administrador en caso de error. 		
Restricciones			
<ul style="list-style-type: none"> • Solo se podrán asociar proyectos con clientes previamente registrados en el sistema. • Cada proyecto debe estar vinculado obligatoriamente a un contrato válido. • No se podrán asociar entregables sin haber definido previamente el proyecto al que pertenecen. • Los entregables deben registrarse con fechas válidas y no pueden estar duplicados dentro del mismo proyecto. • El sistema no debe permitir asociaciones múltiples que generen inconsistencias (por ejemplo, un proyecto ligado a varios contratos activos). 			

HISTORIA DE USUARIO			
Prioridad: Alta			
CÓDIGO DEL REQUERIMIENTO:	RF12	Actor	Administrador
NOMBRE DEL REQUERIMIENTO	Cambiar el estado del proyecto (activo, pausado, finalizado, cancelado)		
Descripción			
Como administrador, quiero poder cambiar el estado de un proyecto en el sistema a activo, pausado, finalizado o cancelado, con el fin de reflejar su situación actual y asegurar un adecuado seguimiento en la gestión de los proyectos.			
Funcionalidad			
El sistema debe permitir al administrador modificar el estado de un proyecto en cualquier momento de su ciclo de vida, de acuerdo con su situación real. Al crear un proyecto, este debe registrarse con un estado inicial definido, y posteriormente el administrador podrá actualizarlo según corresponda. El cambio de estado debe ser inmediato y reflejarse en toda la plataforma, garantizando que los usuarios cuenten siempre con información confiable y actualizada. Además, el sistema debe validar las transiciones de estado, evitando inconsistencias como marcar un proyecto finalizado y luego regresarlo a activo, o cancelar proyectos que ya fueron finalizados. Cada modificación debe registrarse en el historial del proyecto para mantener trazabilidad y control de los cambios realizados.			
Criterios de aceptación	<ol style="list-style-type: none"> 1. El sistema debe permitir al administrador cambiar el estado de un proyecto a activo, pausado, finalizado o cancelado. 2. El sistema debe validar que las transiciones de estado sean correctas y coherentes con el flujo de un proyecto. 3. El sistema debe actualizar de inmediato la información del proyecto en la base de datos y mostrar el estado actual en las consultas y listados. 4. El sistema debe registrar el cambio de estado en un historial asociado al proyecto. 5. El sistema debe mostrar un mensaje de confirmación cuando el cambio se realice exitosamente o notificar en caso de error. 		
Restricciones			
<ul style="list-style-type: none"> • Un proyecto finalizado no puede volver a estados anteriores (activo o pausado). • Un proyecto cancelado no puede reactivarse. • El sistema no debe permitir cambios de estado si el proyecto no cuenta con un cliente y contrato asociado previamente. • Solo el administrador tiene permisos para cambiar el estado de un proyecto. • Los cambios de estado deben registrarse con fecha y hora para garantizar trazabilidad. 			

HISTORIA DE USUARIO			
Prioridad: Alta			
CÓDIGO DEL REQUERIMIENTO:	RF13	Actor	Administrador
NOMBRE DEL REQUERIMIENTO	Registrar avances de proyectos		
Descripción			
Como administrador, quiero poder cambiar el estado de un proyecto en el sistema a activo, pausado, finalizado o cancelado, con el fin de reflejar su situación actual y asegurar un adecuado seguimiento en la gestión de los proyectos.			
Funcionalidad			
El sistema debe permitir al administrador ingresar registros de avances asociados a cada proyecto, especificando detalles como la fecha del avance, la descripción de las actividades realizadas, el porcentaje de progreso alcanzado y el responsable que reporta la información. Estos avances deben quedar vinculados directamente al proyecto correspondiente y visibles en su historial, de manera que se pueda consultar en cualquier momento la evolución del mismo. Asimismo, el sistema debe permitir agregar múltiples avances a lo largo del ciclo de vida de un proyecto, manteniendo la información ordenada cronológicamente y garantizando que los datos registrados no se pierdan ni se sobrescriban. Esta funcionalidad permitirá llevar un control más preciso y transparente del estado real de cada proyecto.			
Criterios de aceptación	<ol style="list-style-type: none"> 1. El sistema debe permitir registrar avances con información obligatoria: fecha, descripción, porcentaje de progreso y responsable. 2. El sistema debe asociar cada avance al proyecto correspondiente de forma automática. 3. El sistema debe mostrar los avances en orden cronológico dentro de la vista del proyecto. 4. El sistema debe permitir consultar el historial completo de avances de un proyecto. 5. El sistema debe mostrar un mensaje de confirmación al registrar un avance exitosamente o notificar en caso de error. 		
Restricciones			
<ul style="list-style-type: none"> • Solo se pueden registrar avances en proyectos que estén en estado activo o pausado. • El porcentaje de progreso debe ser un valor numérico entre 0% y 100%. • No se debe permitir registrar dos avances con la misma fecha y descripción exacta en un mismo proyecto. • Solo los administradores o responsables asignados podrán registrar avances en los proyectos. • Una vez registrado un avance, este no puede eliminarse, solo corregirse o actualizarse con un nuevo registro. 			

HISTORIA DE USUARIO			
Prioridad: Alta			
CÓDIGO DEL REQUERIMIENTO:	RF14	Actor	Administrador
NOMBRE DEL REQUERIMIENTO	Generar un contrato asociado a un proyecto		
Descripción			
Como administrador, quiero que el sistema permita generar un contrato asociado a un proyecto, con el fin de formalizar el acuerdo con el cliente y dejar documentados los compromisos, plazos y condiciones de ejecución del mismo.			
Funcionalidad			
El sistema debe permitir al administrador generar un contrato directamente vinculado a un proyecto específico, tomando como base la información previamente registrada del cliente, la propuesta aceptada y los detalles del proyecto. Al momento de la creación, el contrato debe incluir datos esenciales como la identificación del cliente, el alcance del proyecto, los entregables comprometidos, los plazos de ejecución y las condiciones económicas. Una vez generado, el contrato quedará registrado en el sistema y asociado al proyecto de manera única, garantizando que no existan duplicados y que cada proyecto cuente con su contrato formal. Además, el sistema debe permitir la consulta y descarga del contrato, asegurando que se pueda acceder al documento en cualquier momento durante el ciclo de vida del proyecto.			
Criterios de aceptación	<ol style="list-style-type: none"> 1. El sistema debe permitir generar un contrato solo si el proyecto ya está registrado y cuenta con cliente y propuesta asociada. 2. El contrato debe contener información obligatoria: cliente, proyecto, alcance, entregables, fechas y condiciones económicas. 3. Cada proyecto debe tener un único contrato válido asociado. 4. El sistema debe guardar el contrato en la base de datos y asociarlo al proyecto automáticamente. 5. El sistema debe permitir la visualización y descarga del contrato generado. 6. El sistema debe mostrar un mensaje de confirmación cuando el contrato se genere correctamente o un error si la acción no se completa. 		
Restricciones			
<ul style="list-style-type: none"> • No se puede generar un contrato para proyectos que estén cancelados o finalizados. • No se debe permitir la creación de más de un contrato activo por proyecto. • El contrato debe estar vinculado de forma obligatoria al cliente y al proyecto correspondientes. • Solo los administradores tienen permisos para generar contratos. • Los datos del contrato no pueden ser modificados una vez generado; en caso de cambios, debe crearse un nuevo contrato con versión actualizada. 			

HISTORIA DE USUARIO			
Prioridad: Alta			
CÓDIGO DEL REQUERIMIENTO:	RF15	Actor	Administrador
NOMBRE DEL REQUERIMIENTO	Generar un contrato asociado a un proyecto		
Descripción			
Como administrador, quiero registrar en el sistema las condiciones del proyecto, sus fechas de inicio y finalización, así como el valor total, con el fin de garantizar que cada proyecto cuente con información clara y detallada que facilite su seguimiento, control financiero y cumplimiento de plazos.			
Funcionalidad			
El sistema debe permitir al administrador registrar las condiciones generales de un proyecto, definiendo los términos acordados con el cliente, los criterios de cumplimiento y las observaciones necesarias. Además, debe permitir establecer las fechas de inicio y finalización, validando que la fecha de inicio no sea posterior a la de fin y que ambas sean coherentes con el cronograma del proyecto. De igual forma, el sistema debe permitir ingresar el valor total del proyecto, asegurando que se registre como un dato numérico válido y positivo. Esta información debe quedar asociada directamente al proyecto en la base de datos y reflejarse en sus reportes, de manera que se pueda consultar fácilmente durante la gestión y el seguimiento del mismo.que facilite su seguimiento, control financiero y cumplimiento de plazos.			
Criterios de aceptación	<ol style="list-style-type: none"> 1. El sistema debe permitir registrar condiciones generales del proyecto en un campo de texto. 2. El sistema debe validar que la fecha de inicio sea anterior o igual a la fecha de finalización. 3. El sistema debe permitir registrar el valor total del proyecto como un número positivo. 4. La información registrada debe guardarse y asociarse al proyecto de manera automática. 5. El sistema debe mostrar mensajes de confirmación cuando los datos se guarden correctamente o de error si el registro no se completa 		
Restricciones			
<ul style="list-style-type: none"> • No se pueden registrar fechas de inicio o fin vacías o con un formato incorrecto. • El valor total no puede ser negativo ni contener caracteres no numéricos. • No se podrán registrar condiciones, fechas o valores si el proyecto no tiene un cliente y contrato asociados previamente. • Una vez definido el valor total, este no podrá modificarse directamente; en caso de ajustes se deberá registrar una actualización mediante una versión o adenda. • Solo los administradores tienen permisos para realizar este registro. 			

HISTORIA DE USUARIO			
Prioridad: Alta			
CÓDIGO DEL REQUERIMIENTO:	RF16	Actor	Administrador
NOMBRE DEL REQUERIMIENTO	consultar contratos asociados a proyectos		
Descripción			
Como administrador, quiero consultar los contratos asociados a un proyecto, con el fin de acceder fácilmente a la información legal y documental que respalda su ejecución.			
Funcionalidad			
El sistema debe permitir visualizar los contratos vinculados a un proyecto, mostrando información relevante como cliente, condiciones, fechas, valor y estado, garantizando que el administrador tenga acceso rápido y centralizado a la documentación.			
Criterios de aceptación	<ol style="list-style-type: none"> 1. El sistema debe mostrar los contratos asociados a un proyecto de forma clara y organizada. 2. El sistema debe permitir visualizar información clave: cliente, condiciones, fechas, valor y estado del contrato. 3. El sistema debe validar que solo se muestren contratos correspondientes al proyecto seleccionado. 4. El sistema debe notificar al administrador en caso de que el proyecto no tenga contratos asociados. 5. El sistema debe permitir la descarga o consulta detallada del contrato. 		
Restricciones			
<ul style="list-style-type: none"> • Solo se podrán consultar contratos de proyectos existentes en el sistema. • El administrador debe tener permisos para acceder a la información del contrato. • No se deben mostrar contratos inactivos o cancelados como si estuvieran vigentes. • Los contratos deben mantenerse asociados únicamente al proyecto correspondiente para evitar confusión. 			

HISTORIA DE USUARIO			
Prioridad: Alta			
CÓDIGO DEL REQUERIMIENTO:	RF17	Actor	Administrador
NOMBRE DEL REQUERIMIENTO	Registrar entregables con fecha límite y estado (pendiente, entregado, aprobado)		
Descripción			
Como administrador, quiero registrar entregables dentro de un proyecto con fecha límite y estado, con el fin de dar seguimiento a los compromisos establecidos y garantizar el cumplimiento de los objetivos acordados con el cliente.			
Funcionalidad			
El sistema debe permitir registrar entregables asociados a un proyecto, definiendo su descripción, fecha límite y estado (pendiente, entregado, aprobado). Esta información debe quedar vinculada al proyecto y disponible para consulta y seguimiento.			
Criterios de aceptación	<ol style="list-style-type: none"> 1. El sistema debe permitir registrar entregables con descripción, fecha límite y estado inicial. 2. El sistema debe validar que la fecha límite sea una fecha válida y posterior a la de inicio del proyecto. 3. El sistema debe mostrar los entregables organizados por estado y fecha límite. 4. El sistema debe permitir modificar el estado de un entregable a medida que avance su cumplimiento. 5. El sistema debe notificar al administrador si un entregable está próximo a vencer o si ya venció. 		
Restricciones			
<ul style="list-style-type: none"> • No se pueden registrar entregables sin asociarlos a un proyecto existente. • El estado de los entregables debe estar limitado a pendiente, entregado o aprobado. • No se deben permitir registros duplicados con la misma descripción y fecha límite dentro de un mismo proyecto. • Solo los administradores o responsables asignados pueden registrar y actualizar entregables. • Una vez aprobado, un entregable no podrá volver a estados anteriores. 			

HISTORIA DE USUARIO			
Prioridad: Alta			
CÓDIGO DEL REQUERIMIENTO:	RF18	Actor	Administrador
NOMBRE DEL REQUERIMIENTO	Actualizar o eliminar entregables		
Descripción			
Como administrador, quiero actualizar o eliminar los entregables de un proyecto, con el fin de mantener la información al día y eliminar aquellos que ya no sean necesarios o estén duplicados.			
Funcionalidad			
El sistema debe permitir modificar los datos de un entregable, como la descripción, la fecha límite o el estado, y también debe ofrecer la opción de eliminar entregables que no sean relevantes, garantizando que la información registrada sea precisa y actualizada.			
Criterios de aceptación	<ol style="list-style-type: none"> 1. El sistema debe permitir actualizar la descripción, fecha límite y estado de un entregable. 2. El sistema debe guardar los cambios de inmediato y reflejarlos en el proyecto correspondiente. 3. El sistema debe permitir eliminar un entregable y notificar al administrador cuando la acción se complete con éxito. 4. El sistema debe impedir la eliminación de entregables ya aprobados. 5. El sistema debe mostrar mensajes claros de confirmación o error al actualizar o eliminar. 		
Restricciones			
<ul style="list-style-type: none"> • Solo se pueden actualizar o eliminar entregables que pertenezcan a proyectos activos o pausados. • Los entregables aprobados no podrán eliminarse ni volver a estados anteriores. • No se deben permitir modificaciones que dejen campos vacíos o fechas inválidas. • Solo los administradores o responsables autorizados podrán realizar estas operaciones. • Los entregables eliminados deben marcarse como inactivos en la base de datos para mantener trazabilidad. 			

HISTORIA DE USUARIO			
Prioridad: Alta			
CÓDIGO DEL REQUERIMIENTO:	RF19	Actor	Administrador
NOMBRE DEL REQUERIMIENTO	Controlar los cambios de estado de entregables con posibilidad de rollback mediante transacciones		
Descripción			
Como administrador, quiero que el sistema controle los cambios de estado de los entregables y permita revertirlos mediante transacciones, con el fin de garantizar la consistencia de la información y corregir errores en caso de modificaciones incorrectas			
Funcionalidad			
El sistema debe permitir cambiar el estado de un entregable (pendiente, entregado, aprobado) de forma controlada, registrando cada transición en una transacción. En caso de error o inconsistencia, el sistema debe permitir revertir el cambio (rollback), asegurando que la información se mantenga íntegra y confiable.			
Criterios de aceptación	<ol style="list-style-type: none"> 1. El sistema debe registrar cada cambio de estado de un entregable como una transacción. 2. El sistema debe permitir realizar rollback si ocurre un error durante el cambio de estado. 3. El sistema debe impedir cambios de estado inválidos o que rompan la secuencia lógica. 4. El sistema debe guardar un historial de cambios de estado para cada entregable. 5. El sistema debe notificar al administrador en caso de éxito o fallo durante la operación. 		
Restricciones			
<ul style="list-style-type: none"> • Solo los administradores o responsables autorizados podrán cambiar el estado de un entregable. • No se podrá realizar rollback si el entregable ya está en estado aprobado y confirmado. • Los cambios de estado deben seguir un flujo coherente (pendiente → entregado → aprobado). • El sistema debe garantizar atomicidad: el cambio solo se aplicará si toda la transacción es válida. • Los registros de rollback deben quedar documentados para trazabilidad. 			

HISTORIA DE USUARIO			
Prioridad: Alta			
CÓDIGO DEL REQUERIMIENTO:	RF20	Actor	Administrador
NOMBRE DEL REQUERIMIENTO	Registrar ingresos y egresos asociados a proyectos		
Descripción			
Como administrador o contador, quiero registrar los ingresos y egresos asociados a cada proyecto, con el fin de llevar un control financiero que permita evaluar la rentabilidad y el estado económico de los proyectos.			
Funcionalidad			
El sistema debe permitir ingresar, listar, actualizar y eliminar registros financieros de un proyecto, clasificándolos como ingresos o egresos. Cada registro debe incluir fecha, concepto, monto y responsable, y estar vinculado al proyecto correspondiente.			
Criterios de aceptación	<ol style="list-style-type: none"> 1. El sistema debe permitir registrar ingresos y egresos diferenciados por tipo. 2. Cada registro debe incluir proyecto, fecha, concepto, monto y responsable. 3. El sistema debe permitir consultar el historial financiero de un proyecto. 4. El sistema debe permitir actualizar o eliminar un registro existente. 5. El sistema debe generar un balance general (total ingresos total egresos) por proyecto. 		
Restricciones			
<ul style="list-style-type: none"> • Solo administradores o contadores autorizados pueden registrar movimientos financieros. • No se permite ingresar registros sin asociarlos a un proyecto. • Los registros no pueden tener montos negativos. • Una vez cerrado un proyecto, no se podrán agregar nuevos ingresos o egresos. • Toda modificación debe quedar registrada en el historial financiero para trazabilidad. 			

HISTORIA DE USUARIO			
Prioridad: Alta			
CÓDIGO DEL REQUERIMIENTO:	RF21	Actor	Administrador
NOMBRE DEL REQUERIMIENTO	Consultar balance financiero por cliente o rango de fechas		
Descripción			
<p>Como administrador o contador, quiero consultar el balance financiero de los proyectos asociados a un cliente o dentro de un rango de fechas, con el fin de analizar ingresos, egresos y rentabilidad de manera consolidada.</p>			
Funcionalidad			
<p>El sistema debe permitir consultar balances financieros aplicando filtros por cliente y/o rango de fechas. El balance debe mostrar ingresos, egresos y resultado neto (ingresos – egresos), además de permitir visualizar el detalle de los proyectos incluidos.</p>			
Criterios de aceptación	<ol style="list-style-type: none"> 1. El sistema debe mostrar ingresos, egresos y balance neto agrupados por cliente o rango de fechas. 2. Debe permitir aplicar filtros combinados (cliente y fechas) o individuales. 3. El sistema debe mostrar el detalle de proyectos que conforman el balance. 4. El balance debe actualizarse en tiempo real según los registros financieros existentes. 5. El sistema debe permitir exportar el resultado en al menos un formato descargable (ejemplo: PDF o Excel). 		
Restricciones			
<ul style="list-style-type: none"> • Solo usuarios con permisos de administración o contabilidad pueden acceder al balance financiero. • No se deben incluir proyectos eliminados o sin movimientos financieros registrados. • El rango de fechas no puede ser inválido (fecha fin menor a fecha inicio). • El balance debe garantizar integridad y exactitud de los datos, sin permitir alteraciones al exportar. • El sistema debe procesar la información de forma eficiente para no afectar el rendimiento. 			

HISTORIA DE USUARIO			
Prioridad: Alta			
CÓDIGO DEL REQUERIMIENTO:	RF22	Actor	Administrador
NOMBRE DEL REQUERIMIENTO	Usar transacciones para registrar pagos y evitar inconsistencias		
Descripción			
El sistema debe ejecutar el registro de pagos dentro de una transacción, asegurando que todos los pasos (registro del pago, actualización del balance del proyecto y trazabilidad) se completen correctamente o se reviertan en caso de error.			
Funcionalidad			
El sistema debe ejecutar el registro de pagos dentro de una transacción, asegurando que todos los pasos (registro del pago, actualización del balance del proyecto y trazabilidad) se completen correctamente o se reviertan en caso de error.			
Criterios de aceptación		<ol style="list-style-type: none"> 1. El sistema debe registrar cada pago como parte de una transacción atómica. 2. Si ocurre un error en cualquiera de los pasos, la transacción debe revertirse y no guardar datos parciales. 3. El sistema debe actualizar automáticamente el balance financiero del proyecto al completar el pago. 4. Cada transacción debe quedar registrada en un historial para trazabilidad. 5. El sistema debe notificar al usuario si una transacción falla o se completa exitosamente. 	
Restricciones			
<ul style="list-style-type: none"> • Solo usuarios autorizados (administradores o contadores) pueden registrar pagos. • No se deben permitir pagos sin estar asociados a un proyecto válido. • El monto del pago no puede ser negativo ni exceder los valores definidos en el contrato. • El sistema debe garantizar la consistencia incluso en escenarios de concurrencia (múltiples usuarios registrando pagos al mismo tiempo). • Las transacciones deben seguir los principios ACID (Atomicidad, Consistencia, Aislamiento, Durabilidad). 			

HISTORIA DE USUARIO			
Prioridad: Alta			
CÓDIGO DEL REQUERIMIENTO:	RF23	Actor	Administrador
NOMBRE DEL REQUERIMIENTO	Menús interactivos para acceder a cada módulo con mensajes claros en consola		
Descripción			
<p>Como usuario del sistema, quiero contar con menús interactivos en consola para acceder de manera organizada a los diferentes módulos (clientes, propuestas, proyectos, contratos, entregables y finanzas), con mensajes de error, advertencia y confirmación claros, para facilitar la navegación y reducir errores al interactuar con el sistema.</p>			
Funcionalidad			
<p>El sistema debe mostrar un menú principal que permita acceder a los módulos, y cada módulo debe tener su propio submenú con las opciones correspondientes. Además, el sistema debe desplegar mensajes claros al usuario en consola al confirmar acciones, advertir sobre riesgos (como eliminaciones) o notificar errores en los datos ingresados.</p>			
Criterios de aceptación	<ol style="list-style-type: none"> 1. El sistema debe mostrar un menú principal con acceso a: clientes, propuestas, proyectos, contratos, entregables y finanzas. 2. Cada módulo debe tener opciones claras y numeradas en consola. 3. El sistema debe mostrar mensajes de confirmación al realizar acciones exitosas. 4. El sistema debe mostrar mensajes de advertencia antes de acciones críticas (ejemplo: eliminar un registro). 5. El sistema debe mostrar mensajes de error cuando el usuario ingrese información inválida o incompleta. 		
Restricciones			
<ul style="list-style-type: none"> • El sistema debe validar la entrada del usuario (solo números para menús, datos requeridos completos). • Los mensajes deben ser claros y consistentes en todo el sistema. • No se debe permitir salir de un módulo sin confirmar la acción de retorno al menú principal. • El menú debe funcionar en entornos de consola sin necesidad de interfaces gráficas. • El diseño del menú debe ser flexible para permitir agregar nuevos módulos en el futuro. 			

HISTORIA DE USUARIO			
Prioridad: Alta			
CÓDIGO DEL REQUERIMIENTO:	RF24	Actor	Administrador
NOMBRE DEL REQUERIMIENTO	Interfaz de consola intuitiva, menús interactivos y mensajes claros con rendimiento óptimo		
Descripción			
<p>Como usuario del sistema, quiero que la aplicación ofrezca una interfaz de consola intuitiva y fácil de usar, con menús interactivos bien organizados y mensajes claros, para que la navegación sea ágil, las operaciones comprensibles y el sistema responda con buen rendimiento incluso al manejar grandes volúmenes de datos.</p>			
Funcionalidad			
<p>El sistema debe mostrar menús interactivos, jerárquicos y accesibles desde consola, con mensajes de confirmación, advertencia y error consistentes. Además, debe responder en tiempos óptimos al realizar consultas, registros y actualizaciones, garantizando una experiencia fluida para el usuario.</p>			
Criterios de aceptación		<ol style="list-style-type: none"> 1. El menú principal debe ofrecer acceso directo a todos los módulos: clientes, propuestas, proyectos, contratos, entregables y finanzas. 2. Los submenús deben estar organizados y permitir volver fácilmente al menú principal. 3. Los mensajes en consola deben ser breves, claros y coherentes en todo el sistema. 4. El tiempo de respuesta para operaciones comunes (crear, listar, actualizar, eliminar) no debe superar los 2 segundos en condiciones normales de uso. 5. El sistema debe manejar hasta 500 registros activos por módulo sin afectar el rendimiento. 	
Restricciones			
<ul style="list-style-type: none"> • La interfaz debe ser exclusivamente en consola, sin necesidad de interfaz gráfica. • Los mensajes deben estar normalizados (ejemplo: siempre iniciar con "Éxito:", "Error:", "Advertencia:"). • No se deben permitir bloqueos en la navegación entre menús. • El diseño debe ser escalable para integrar nuevos módulos sin degradar el rendimiento. • El sistema debe optimizar operaciones de lectura y escritura para mantener la fluidez. 			

HISTORIA DE USUARIO			
Prioridad: Alta			
CÓDIGO DEL REQUERIMIENTO:	RF25	Actor	Administrador
NOMBRE DEL REQUERIMIENTO	Ejecución de tareas en menos de 3 segundos		
Descripción			
<p>Como usuario del sistema, quiero que al ejecutar tareas como consultas, creación o actualización de datos, la aplicación responda en menos de 3 segundos en condiciones normales de operación, para trabajar de forma ágil y sin retrasos.</p>			
Funcionalidad			
<p>El sistema debe garantizar tiempos de respuesta óptimos para las operaciones principales (consultar, crear, actualizar, eliminar registros), asegurando que los procesos internos estén optimizados y que no se generen esperas innecesarias en consola.</p>			
Criterios de aceptación	<ol style="list-style-type: none"> 1. Todas las operaciones CRUD (crear, leer, actualizar, eliminar) deben ejecutarse en menos de 3 segundos bajo condiciones normales. 2. Las consultas deben retornar resultados de forma inmediata cuando se realicen sobre menos de 1000 registros. 3. El sistema debe mostrar un mensaje de progreso si alguna operación excepcional tarda más del tiempo esperado. 4. No se deben producir bloqueos o congelamientos de la aplicación durante las operaciones. 5. El rendimiento debe mantenerse estable aunque varios usuarios realicen operaciones de manera concurrente 		
Restricciones			
<ul style="list-style-type: none"> • Las pruebas de rendimiento deben hacerse con una base de datos que contenga al menos 500 registros activos por módulo. • El tiempo de respuesta máximo de 3 segundos aplica en condiciones normales de red y hardware estándar. • Operaciones más complejas (reportes financieros, consultas avanzadas) pueden superar este límite, pero nunca deben exceder los 5 segundos. • El sistema debe estar optimizado para escalar en volumen de datos sin perder estabilidad. 			

HISTORIA DE USUARIO			
Prioridad: Alta			
CÓDIGO DEL REQUERIMIENTO:	RF26	Actor	Administrador
NOMBRE DEL REQUERIMIENTO	Validación de registros duplicados o incompletos		
Descripción			
Como usuario del sistema, quiero que los registros no puedan guardarse duplicados ni con información incompleta, para asegurar la calidad y confiabilidad de los datos almacenados.			
Funcionalidad			
El sistema debe validar que cada registro sea único y que todos los campos obligatorios estén completos antes de permitir su almacenamiento en la base de datos.			
Criterios de aceptación	<ol style="list-style-type: none"> 1. El sistema debe impedir la creación de registros duplicados en entidades clave (clientes, proyectos, contratos, propuestas, entregables). 2. Los campos obligatorios deben ser validados antes de guardar cualquier registro. 3. Si se intenta guardar un registro incompleto, el sistema debe mostrar un mensaje de error indicando los campos faltantes. 4. Al detectar un registro duplicado, se debe notificar al usuario con un mensaje de advertencia claro. 5. El sistema debe permitir la actualización de registros existentes sin generar duplicados. 		
Restricciones			
<ul style="list-style-type: none"> • La verificación de duplicados debe realizarse sobre campos únicos como identificaciones, correos electrónicos, códigos de contrato o número de proyecto. • Ningún registro debe almacenarse si falta al menos un campo marcado como obligatorio. • El sistema debe ser consistente en la validación tanto en la creación como en la actualización de registros. • La validación debe ejecutarse en menos de 2 segundos para no afectar el rendimiento. 			

HISTORIA DE USUARIO			
Prioridad: Alta			
CÓDIGO DEL REQUERIMIENTO:	RF27	Actor	Administrador
NOMBRE DEL REQUERIMIENTO	Uso de transacciones en operaciones críticas		
Descripción			
Como usuario del sistema, quiero que las operaciones críticas como el registro de pagos o la actualización de entregables se realicen dentro de transacciones en MongoDB, para asegurar la consistencia e integridad de los datos en caso de errores o fallos durante el proceso.			
Funcionalidad			
El sistema debe ejecutar operaciones críticas utilizando transacciones en MongoDB, de forma que todos los cambios se apliquen únicamente si la transacción se completa con éxito; en caso contrario, se deben revertir los cambios realizados para evitar inconsistencias.			
Criterios de aceptación	<ol style="list-style-type: none"> 1. Todas las operaciones críticas deben ejecutarse dentro de una transacción (ej. pagos, entregables, contratos, finanzas). 2. Si ocurre un error durante la transacción, el sistema debe hacer un rollback automático para revertir los cambios. 3. El usuario debe recibir un mensaje claro indicando si la operación fue confirmada o revertida. 4. Las transacciones deben garantizar la atomicidad: o se aplican todos los cambios, o no se aplica ninguno. 5. Las pruebas deben demostrar que no se producen inconsistencias de datos incluso ante fallos de red o caídas del sistema. 		
Restricciones			
<ul style="list-style-type: none"> • Las nuevas funcionalidades deben añadirse mediante extensiones o clases adicionales, no alterando directamente el código central. • Los patrones de diseño deben seleccionarse según la naturaleza del módulo a implementar. • Se debe garantizar compatibilidad con las versiones previas del sistema al incorporar nuevas funciones. • La arquitectura debe soportar pruebas unitarias y de integración sin afectar módulos ya implementados. 			

HISTORIA DE USUARIO			
Prioridad: Alta			
CÓDIGO DEL REQUERIMIENTO:	RF28	Actor	Administrador
NOMBRE DEL REQUERIMIENTO	Arquitectura extensible con principios SOLID y patrones de diseño		
Descripción			
<p>Como desarrollador, quiero que la arquitectura de la aplicación permita agregar nuevas funcionalidades sin necesidad de modificar drásticamente el código existente, aplicando principios SOLID y patrones de diseño, con el fin de garantizar mantenibilidad, escalabilidad y facilidad de evolución del sistema.</p>			
Funcionalidad			
<p>El sistema debe estar diseñado bajo una arquitectura modular y desacoplada, implementando principios SOLID y patrones de diseño, de manera que las nuevas funciones se puedan integrar mediante extensiones o módulos sin afectar el núcleo del sistema.</p>			
Criterios de aceptación	<ol style="list-style-type: none"> 1. La arquitectura debe permitir agregar módulos (ej. clientes, proyectos, finanzas) sin cambios drásticos en el código base. 2. El diseño debe cumplir con los principios SOLID (responsabilidad única, abierto/cerrado, sustitución de Liskov, segregación de interfaces, inversión de dependencias). 3. Se deben aplicar patrones de diseño adecuados (ej. Factory, Repository, Observer, Strategy) para desacoplar responsabilidades. 4. El sistema debe mantener un bajo acoplamiento y alta cohesión en sus módulos. 5. La documentación del código debe reflejar las decisiones arquitectónicas implementadas. 		
Restricciones			
<ul style="list-style-type: none"> • Las nuevas funcionalidades deben añadirse mediante extensiones o clases adicionales, no alterando directamente el código central. • Los patrones de diseño deben seleccionarse según la naturaleza del módulo a implementar. • Se debe garantizar compatibilidad con las versiones previas del sistema al incorporar nuevas funciones. • La arquitectura debe soportar pruebas unitarias y de integración sin afectar módulos ya implementados. 			

HISTORIA DE USUARIO			
Prioridad: Alta			
CÓDIGO DEL REQUERIMIENTO:	RF29	Actor	Administrador
NOMBRE DEL REQUERIMIENTO	Manejo de errores, validaciones y excepciones		
Descripción			
<p>Como usuario y administrador del sistema, quiero que la aplicación maneje de forma adecuada los errores de conexión, las validaciones de datos y las excepciones durante la ejecución, con el fin de evitar caídas inesperadas y garantizar una experiencia confiable y estable.</p>			
Funcionalidad			
<p>El sistema debe contar con mecanismos de manejo de errores y validaciones que permitan capturar excepciones, mostrar mensajes claros y registrar incidencias sin interrumpir la ejecución de la aplicación.</p>			
Criterios de aceptación		<ol style="list-style-type: none"> 1. Los errores de conexión a la base de datos deben ser detectados y notificados con mensajes claros al usuario. 2. Los datos ingresados deben validarse antes de ser procesados (ej. campos obligatorios, formatos válidos, rangos permitidos). 3. Todas las excepciones deben ser capturadas y gestionadas, evitando que el sistema se cierre de manera inesperada. 4. Los mensajes de error deben ser descriptivos pero no revelar información sensible. 5. Se debe registrar en un log interno toda excepción crítica para análisis y soporte. 	
Restricciones			
<ul style="list-style-type: none"> • No se permitirá la inserción de datos incompletos o inválidos. • Los mensajes de error deben diferenciar entre advertencia, error crítico y error de validación. • La aplicación no debe detenerse ante una excepción, salvo que sea un fallo crítico no recuperable. • El sistema debe soportar la reconexión automática a la base de datos en caso de fallas temporales. 			

5. METODOLOGÍA

➤ Metodología Ágil: Scrum

El desarrollo de esta aplicación de línea de comandos (CLI) para la gestión del portafolio profesional se ha llevado a cabo utilizando la metodología ágil Scrum, la cual permite una gestión flexible, iterativa y colaborativa del proyecto.

- Roles definidos: Se asignaron los roles de Scrum Master, Product Owner y Equipo de Desarrollo para facilitar la coordinación y comunicación efectiva durante todo el ciclo de vida del proyecto.
- Planificación por sprints: El trabajo se organizó en ciclos cortos denominados sprints, donde se planifican y desarrollan funcionalidades específicas priorizadas en el Product Backlog.
- Gestión de requerimientos: El Product Backlog contiene la lista priorizada de todas las funcionalidades, mejoras y correcciones, que se revisan y actualizan continuamente.
- Reuniones regulares: Se realizaron reuniones de planificación, seguimiento diario (Daily Scrum), revisión de sprint y retrospectiva, para asegurar la adaptación constante y la mejora del proceso.
- Entrega incremental: Cada sprint buscó entregar incrementos funcionales y verificables que aportaran valor al usuario final, garantizando la calidad y satisfacción del cliente.

➤ Buenas Prácticas Técnicas con [Node.js](#)

Para el desarrollo técnico de la aplicación se siguieron buenas prácticas modernas, enfocadas en mantener un código organizado, reutilizable y eficiente:

- Programación Orientada a Objetos (POO): Se utilizó para definir claramente los modelos de datos y la lógica del negocio, aplicando principios SOLID para facilitar la mantenibilidad y extensibilidad del código.

- Patrones de diseño: Se aplicaron patrones como Factory para la creación de objetos y Command para estructurar los comandos de la CLI, promoviendo un código modular y desacoplado.
- Modularidad: El código se organizó en módulos y carpetas específicas para models, servicios, comandos y utilidades, facilitando la navegación y el trabajo en equipo.
- Programación asíncrona: Se manejaron operaciones de entrada/salida y conexión a la base de datos MongoDB mediante async/await, garantizando un flujo eficiente y no bloqueante.
- Gestión de dependencias: Se empleó NPM para la administración de librerías esenciales como commander, inquirer, chalk, dotenv y el driver oficial de MongoDB, asegurando facilidad de instalación y actualización.

Esta metodología combinada con buenas prácticas técnicas permitió desarrollar un software robusto, escalable y alineado con las necesidades del usuario, favoreciendo la productividad y calidad del proyecto.

CONCLUSIONES

En este proyecto se diseñó y desarrolló una aplicación de línea de comandos (CLI) en Node.js destinada a mejorar la gestión y organización del portafolio profesional de una empresa de freelancers. La herramienta logró centralizar y automatizar la administración de clientes, propuestas, proyectos, contratos, entregables y finanzas, resolviendo problemas previos de dispersión y duplicidad de información.

La implementación basada en la metodología ágil Scrum permitió una planificación iterativa y colaborativa que facilitó la adaptación continua a los requerimientos del proyecto y la entrega incremental de funcionalidades de alto valor. Además, la adopción de principios SOLID, programación orientada a objetos y patrones de diseño garantizó un código modular, mantenible y escalable.

El uso de MongoDB con transacciones proporcionó una persistencia segura y consistente de la información crítica, especialmente en el manejo de entregables y operaciones financieras. La aplicación CLI ofrece una interfaz eficiente y amigable para usuarios técnicos, optimizando la productividad diaria.

Como resultados, se espera que esta solución impacte positivamente en la organización y control del portafolio freelance, con posibilidades de futuras ampliaciones y mejoras, como integración con sistemas web o móviles y mejoras en la experiencia de usuario.

En resumen, el proyecto demostró que es posible construir sistemas robustos y adaptativos con tecnologías modernas y metodologías ágiles, atendiendo a necesidades reales y facilitando la gestión profesional de freelancers.