

GESTOR DE PORTAFOLIO DE PROYECTOS FREELANCE

JHON ISAAC MEDINA MENDOZA

GRUPO U2

JUAN MARINO

**CAMPUSLANDS
SALON U2
RUTA NODE.JS
FLORIDABLANCA
2025**

GESTOR DE PORTAFOLIO DE PROYECTOS FREELANCE

1. SITUACIÓN PROBLEMA

Los FreeLancer, al gestionar múltiples clientes y proyectos, se enfrentan a diversos desafíos en relación a la organización, el seguimiento y la documentación de su trabajo. En muchos casos, la falta de una herramienta unificada para registrar propuestas, contratos, entregables y movimientos financieros impide una gestión efectiva, impactando la productividad y la profesionalidad.

Este proyecto busca desarrollar una aplicación CLI que centralice todas esas gestiones, ofreciendo una solución digital accesible y escalable que permita al FreeLancer operar de forma más profesional, ordenada y eficiente.

2. LEVANTAMIENTO DE REQUERIMIENTOS

Método empleado para definir los requerimientos

El levantamiento de requerimientos de este proyecto se realizó combinando varias estrategias, alineadas con prácticas profesionales:

1. Entrevistas y consulta con compañeros desarrolladores:
Se indagó con otros programadores sobre los elementos que normalmente gestionan al trabajar por encargo (freelance): clientes, propuestas, entregas, finanzas, etc.
2. Experiencias personales del desarrollador:
Se aprovechó el conocimiento previo de Isaac Medina como estudiante de programación y participante en simulaciones de gestión de proyectos para identificar problemas comunes (falta de control, seguimiento, profesionalismo, etc.).
3. Investigación comparativa:
Se estudiaron otras herramientas o plataformas de gestión de proyectos freelance (Trello, Bonsai, Notion, entre otras), identificando funcionalidades útiles como:
 - Registro de clientes
 - Contratos vinculados a proyectos
 - Control de ingresos/egresos
 - Estados para entregables y propuestas

- Reportes de balance financiero
- 4. Simulación de casos de uso reales:
Se diseñaron historias de usuario partiendo de escenarios típicos de freelancers. Cada historia fue transformada en requerimientos técnicos (RFxx), con sus respectivos criterios de aceptación y validaciones.

Resultado del proceso de levantamiento

- Se identificaron más de 35 requerimientos funcionales y no funcionales, organizados en módulos.
- Se crearon 36 historias de usuario completas, validadas con criterios de aceptación, validaciones, restricciones y pruebas esperadas.
- El resultado ha sido una arquitectura modular CLI, escalable, que refleja necesidades reales del trabajo independiente como desarrollador.

3. REQUERIMIENTOS

3.1. Requerimientos Funcionales

Módulo 1 – Gestión de Clientes

- RF01: Registrar nuevos clientes con nombre, correo, teléfono y tipo (empresa/independiente).
- RF02: Listar todos los clientes existentes.
- RF03: Editar los datos de un cliente existente.
- RF04: Eliminar un cliente, previa validación de relaciones.
- RF05: Volver al menú principal desde el submenú de clientes.

Módulo 2 – Gestión de Propuestas

- **RF06:** Crear propuestas asociadas a un cliente con descripción, tecnologías, tipo (frontend/backend/fullstack), precio, plazos y estado inicial.
- **RF07:** Listar todas las propuestas junto con el nombre del cliente relacionado.
- **RF08:** Cambiar el estado de una propuesta (pendiente, aceptada, rechazada).

- **RF09:** Eliminar propuestas existentes.
- **RF10:** Volver al menú principal desde el submenú de propuestas.

Módulo 3 – Gestión de Proyectos

- **RF11:** Crear un proyecto manualmente, asociándolo a un cliente.
- **RF12:** Listar todos los proyectos existentes.
- **RF13:** Ver proyectos filtrados por cliente.
- **RF14:** Cambiar el estado de un proyecto (activo, pausado, finalizado, cancelado).
- **RF15:** Eliminar un proyecto (con validaciones si tiene contratos o entregables).
- **RF16:** Volver al menú principal desde el submenú de proyectos.

Módulo 4 – Gestión de Contratos

- **RF17:** Crear contrato asociado a un proyecto existente.
- **RF18:** Consultar el contrato de un proyecto específico.
- **RF19:** Listar todos los contratos registrados.
- **RF20:** Volver al menú principal desde el submenú de contratos.

Módulo 5 – Gestión de Entregables

- **RF21:** Crear entregables para un proyecto con nombre, descripción, estado inicial y fecha límite.
- **RF22:** Listar entregables asociados a un proyecto.
- **RF23:** Cambiar el estado de un entregable (pendiente, entregado, aprobado, rechazado).
- **RF24:** Eliminar entregables, con restricciones y confirmación.
- **RF25:** Volver al menú principal desde el submenú de entregables.

Módulo 6 – Gestión Financiera

- **RF26:** Registrar ingresos por proyecto.

- **RF27:** Registrar egresos por proyecto.
- **RF28:** Ver balance financiero por cliente.
- **RF29:** Ver balance financiero por rango de fechas.
- **RF30:** Listar todas las operaciones financieras registradas.
- **RF31:** Volver al menú principal desde el submenú financiero.

Módulo 7 – Validaciones y Seguridad

- **RF32:** Validar todos los campos de entrada (tipo, requeridos, formato, longitud).
- **RF33:** Validar relaciones entre entidades (cliente-propuesta, proyecto-contrato, etc.).
- **RF36:** Prevenir registros duplicados.

Módulo 8 – Experiencia CLI Amigable

- **RF37:** Navegación clara entre menús y submenús.
- **RF38:** Interfaz visual organizada (con chalk, inquirer y estructura modular).
- **RF39:** Mensajes de confirmación, éxito o error con retroalimentación clara.
- **RF40:** Estructura de carpetas organizada por módulo (/utils, /services, /models, etc.).
- **RF41:** Mostrar pausa después de cada acción para lectura.

3.2. Requerimientos No Funcionales

- **RNF01:** La aplicación debe ejecutarse completamente desde consola (CLI).
- **RNF02:** El proyecto debe estar desarrollado con Node.js usando Programación Orientada a Objetos.
- **RNF03:** Aplicar principios SOLID en clases y módulos.
- **RNF04:** Incluir al menos dos patrones de diseño (como Repository, Factory, Command, Observer, etc.).

- **RNF05:** Persistir los datos en MongoDB usando el driver oficial (mongodb), no mongoose.
- **RNF06:** Usar transacciones en operaciones críticas como finanzas o entrega de entregables.
- **RNF07:** Utilizar librerías npm para mejorar la experiencia CLI: inquirer, chalk, dotenv, dayjs, etc.
- **RNF08:** Estructurar el repositorio con carpetas claras: /models, /services, /menus, /config, /utils.
- **RNF09:** Usar convención de commits con formato Conventional Commits (feat:, fix:, docs:, etc.).
- **RNF10:** Incluir un README.md completo con: descripción, instrucciones, estructura, principios SOLID y patrones aplicados.
- **RNF11:** Documentar la planeación Scrum en un PDF entregado en el repositorio.
- **RNF12:** Entregar un video de máximo 7 minutos explicando principios, patrones y demo funcional del sistema.
- **RNF13:** El sistema debe ser modular y escalable, permitiendo agregar nuevos módulos sin reestructurar la base.
- **RNF14:** Todos los errores deben manejarse con retroalimentación amigable, sin mostrar trazas técnicas.
- **RNF15:** El proyecto debe poder ejecutarse con solo node index.js y estar configurado con variables de entorno (.env).

4. HISTORIAS DE USUARIO CON CRITERIOS DE ACEPTACION

HISTORIA DE USUARIO 1			
Prioridad: Alta			
CÓDIGO DEL REQUERIMIENTO:	RF01	Actor	Freelancer
NOMBRE DEL REQUERIMIENTO	Crear nuevo cliente		
Descripción			
Como Freelancer quiero registrar un nuevo cliente con su información básica, para poder asociarlo a futuras propuestas y proyectos.			
Funcionalidad			
Permitir el registro de clientes nuevos solicitando campos obligatorios. El sistema debe validar que no existan duplicados por correo y que el formato de los datos sea correcto. Si el cliente se registra exitosamente, se mostrará un mensaje de confirmación.			
Criterios de aceptación	<ul style="list-style-type: none">El sistema debe solicitar nombre, correo, teléfono y tipo de cliente.Todos los campos deben ser obligatorios.Se debe validar el formato del correo.No deben permitirse correos duplicados.Al registrar exitosamente, debe mostrarse un mensaje de confirmación.		
Restricciones			
No deben permitirse registros duplicados por correo electrónico.			

HISTORIA DE USUARIO 2			
Prioridad: Alta			
CÓDIGO DEL REQUERIMIENTO:	RF02	Actor	Freelancer
NOMBRE DEL REQUERIMIENTO	Listar clientes registrados		
Descripción			
Como Freelancer quiero visualizar una lista de todos mis clientes registrados para revisar su información y poder asociarlos a futuras propuestas o proyectos.			
Funcionalidad			
El sistema debe mostrar todos los clientes existentes en un formato legible, excluyendo datos internos (_id, etc.). La vista debe ser limpia, ordenada y comprensible.			
Criterios de aceptación	<div><input type="checkbox"/> El sistema debe listar todos los clientes registrados.</div> <div><input type="checkbox"/> La vista debe incluir nombre, correo, tipo de cliente y fecha de registro.</div> <div><input type="checkbox"/> Si no hay clientes, se debe mostrar un mensaje como “No hay registros disponibles”.</div>		
Restricciones			
No deben mostrarse campos internos de base de datos.			

HISTORIA DE USUARIO 3			
Prioridad: Alta			
CÓDIGO DEL REQUERIMIENTO:	RF03	Actor	Freelancer
NOMBRE DEL REQUERIMIENTO	Modificar datos de un cliente		
Descripción			
Como Freelancer quiero modificar los datos de un cliente registrado para mantener actualizada su información de contacto.			
Funcionalidad			
Debe permitir seleccionar un cliente existente y editar sus datos (excepto el ID). Todos los datos deben ser validados nuevamente. No se deben aceptar correos duplicados tras la edición.			
Criterios de aceptación	<ul style="list-style-type: none">Se debe permitir seleccionar un cliente para editar.Todos los campos deben validarse nuevamente.No se deben permitir correos duplicados.No debe ser posible modificar el ID.		
Restricciones			
El campo ID no puede ser modificado. El correo debe permanecer único.			

HISTORIA DE USUARIO 4			
Prioridad: Media			
CÓDIGO DEL REQUERIMIENTO:	RF04	Actor	Freelancer
NOMBRE DEL REQUERIMIENTO	Eliminar cliente registrado		
Descripción			
Como Freelancer quiero eliminar un cliente cuando ya no trabaje con él, para mantener actualizada mi lista de contactos.			
Funcionalidad			
El sistema debe permitir eliminar clientes seleccionados, siempre que no tengan proyectos activos. Debe solicitar confirmación previa antes de eliminar.			
Criterios de aceptación	<div><input type="checkbox"/> Se debe permitir seleccionar un cliente para eliminar.</div> <div><input type="checkbox"/> Debe haber una confirmación previa de eliminación.</div> <div><input type="checkbox"/> El sistema debe validar que el cliente no tenga proyectos activos.</div> <div><input type="checkbox"/> Si se confirma, el cliente debe ser eliminado de la base de datos.</div>		
Restricciones			
No se deben eliminar clientes que tengan proyectos activos o en pausa.			

HISTORIA DE USUARIO 5			
Prioridad: Media			
CÓDIGO DEL REQUERIMIENTO:	RF05	Actor	Freelancer
NOMBRE DEL REQUERIMIENTO	Volver al menú principal desde submenús		
Descripción			
Como usuario del sistema quiero poder volver al menú principal desde cualquier submenú para no tener que reiniciar la aplicación al navegar entre módulos.			
Funcionalidad			
Cada submenú (clientes, propuestas, etc.) debe incluir una opción para volver al menú principal. Esta opción debe retornar correctamente sin ejecutar funciones no deseadas ni finalizar la aplicación.			
Criterios de aceptación	<div><input type="checkbox"/> Cada submenú debe incluir la opción “🔙 Volver”.</div> <div><input type="checkbox"/> El sistema debe retornar al menuPrincipal() sin cerrar la aplicación.</div> <div><input type="checkbox"/> No deben ejecutarse funciones accidentales al volver.</div>		
Restricciones			
No debe terminarse el proceso de ejecución ni deben perderse datos.			


HISTORIA DE USUARIO 6			
Prioridad: Alta			
CÓDIGO DEL REQUERIMIENTO:	RF06	Actor	Freelancer
NOMBRE DEL REQUERIMIENTO	Crear nueva propuesta		
Descripción			
Como Freelancer quiero registrar una propuesta para un cliente, indicando el precio, descripción, plazo y tecnologías a usar, para formalizar una posible colaboración.			
Funcionalidad			
El sistema debe permitir la creación de una propuesta asociada a un cliente existente. Se deben ingresar datos clave, validar todos los campos y asignar el estado inicial como "pendiente".			
Criterios de aceptación	<div><input type="checkbox"/> El sistema debe permitir seleccionar un cliente válido.</div> <div><input type="checkbox"/> Se deben ingresar: descripción, precio, plazo en días, tipo de trabajo y tecnologías.</div> <div><input type="checkbox"/> Todos los campos son obligatorios y deben validarse.</div> <div><input type="checkbox"/> El estado inicial de la propuesta debe ser "pendiente".</div>		
Restricciones			
No se debe permitir crear propuestas sin cliente o con campos incompletos.			

HISTORIA DE USUARIO 7			
Prioridad: Alta			
CÓDIGO DEL REQUERIMIENTO:	RF07	Actor	Freelancer
NOMBRE DEL REQUERIMIENTO	Visualizar todas las propuestas registradas		
Descripción			
Como Freelancer quiero poder listar todas las propuestas registradas para tener una vista general de mis negociaciones activas y pasadas.			
Funcionalidad			
El sistema debe listar todas las propuestas existentes en una vista clara que incluya la información relevante del cliente y la propuesta.			
Criterios de aceptación	<div><input type="checkbox"/> El sistema debe mostrar: nombre del cliente, descripción, estado, precio, plazo, fecha de creación y tecnologías.</div> <div><input type="checkbox"/> Si no hay propuestas, debe mostrarse un mensaje como “No hay propuestas registradas”.</div>		
Restricciones			
No deben mostrarse identificadores internos (como _id) en la lista.			

HISTORIA DE USUARIO 8			
Prioridad: Alta			
CÓDIGO DEL REQUERIMIENTO:	RF08	Actor	Freelancer
NOMBRE DEL REQUERIMIENTO	Visualizar propuestas filtradas por cliente		
Descripción			
Como Freelancer quiero ver únicamente las propuestas asociadas a un cliente específico, para facilitar el seguimiento por contacto.			
Funcionalidad			
El sistema debe permitir filtrar las propuestas por cliente y mostrar solo aquellas que estén relacionadas.			
Criterios de aceptación	<div><input type="checkbox"/> El sistema debe permitir seleccionar un cliente.</div> <div><input type="checkbox"/> Debe mostrar: descripción, estado, precio, plazo, fecha y tecnologías.</div> <div><input type="checkbox"/> Si el cliente no tiene propuestas, debe mostrarse un mensaje informativo.</div>		
Restricciones			
El filtrado debe hacerse exclusivamente por clienteld.			

HISTORIA DE USUARIO 9			
Prioridad: Alta			
CÓDIGO DEL REQUERIMIENTO:	RF09	Actor	Freelancer
NOMBRE DEL REQUERIMIENTO	Cambiar el estado de una propuesta		
Descripción			
Como Freelancer quiero actualizar el estado de una propuesta a "aceptada", "rechazada" o mantenerla "pendiente", según el resultado de la negociación.			
Funcionalidad			
El sistema debe permitir seleccionar una propuesta existente y modificar su estado si cumple las condiciones. Al aceptar una propuesta, se debe generar automáticamente un proyecto asociado.			
Criterios de aceptación	<div><input type="checkbox"/> Solo se deben permitir estados válidos: pendiente, aceptada, rechazada.</div> <div><input type="checkbox"/> No puede cambiarse a "aceptada" si ya está aceptada.</div> <div><input type="checkbox"/> Al aceptar, debe generarse automáticamente un proyecto relacionado.</div>		
Restricciones			
No se debe permitir múltiples aceptaciones para la misma propuesta.			

HISTORIA DE USUARIO 10			
Prioridad: Media			
CÓDIGO DEL REQUERIMIENTO:	RF10	Actor	Freelancer
NOMBRE DEL REQUERIMIENTO	Eliminar una propuesta existente		
Descripción			
Como Freelancer quiero poder eliminar propuestas que ya no son relevantes, para mantener organizado mi portafolio de negociaciones.			
Funcionalidad			
El sistema debe permitir eliminar propuestas pendientes o rechazadas. No deben eliminarse propuestas aceptadas.			
Criterios de aceptación	<div><input type="checkbox"/> El sistema debe listar las propuestas eliminables.</div> <div><input type="checkbox"/> Debe haber una confirmación de eliminación.</div> <div><input type="checkbox"/> No se deben eliminar propuestas aceptadas.</div>		
Restricciones			
Las propuestas en estado "aceptada" están protegidas contra eliminación.			

HISTORIA DE USUARIO 11			
Prioridad: Media			
CÓDIGO DEL REQUERIMIENTO:	RF11	Actor	Usuario (CLI)
NOMBRE DEL REQUERIMIENTO	Salir del submenú de propuestas		
Descripción			
Como usuario quiero poder volver al menú principal desde el submenú de propuestas para seguir trabajando en otras secciones de la aplicación.			
Funcionalidad			
El submenú de propuestas debe ofrecer una opción explícita para regresar al menú principal sin cerrar el sistema.			
Criterios de aceptación	<div><input type="checkbox"/> Opción “ Volver” debe estar siempre visible.</div> <div><input type="checkbox"/> Al seleccionar “Volver”, debe retornar al menuPrincipal() correctamente.</div> <div><input type="checkbox"/> El sistema debe conservar el estado de navegación sin errores.</div>		
Restricciones			
No debe invocarse process.exit() en este flujo. No se debe perder información ni ejecutar funciones accidentales.			


HISTORIA DE USUARIO 12			
Prioridad: Alta			
CÓDIGO DEL REQUERIMIENTO:	RF12	Actor	Freelancer
NOMBRE DEL REQUERIMIENTO	Crear nuevo proyecto manualmente		
Descripción			
Como Freelancer quiero registrar un proyecto directamente desde el sistema, sin depender de una propuesta aceptada, para tener mayor flexibilidad.			
Funcionalidad			
El sistema debe permitir crear un nuevo proyecto asociándolo a un cliente existente. Se deben ingresar los datos esenciales del proyecto y validar todos los campos antes de guardar. La fecha de creación debe generarse automáticamente.			
Criterios de aceptación	<div><input type="checkbox"/> El sistema debe permitir seleccionar un cliente existente.</div> <div><input type="checkbox"/> Se deben ingresar nombre, descripción, plazo estimado en días y estado inicial (activo o pausado).</div> <div><input type="checkbox"/> La fecha de creación debe registrarse automáticamente.</div> <div><input type="checkbox"/> Todos los campos deben validarse correctamente.</div>		
Restricciones			
No se deben permitir estados inválidos ni creación sin cliente asociado.			

HISTORIA DE USUARIO 13			
Prioridad: Alta			
CÓDIGO DEL REQUERIMIENTO:	RF13	Actor	Freelancer
NOMBRE DEL REQUERIMIENTO	Visualizar todos los proyectos registrados		
Descripción			
Como Freelancer quiero ver todos mis proyectos activos o inactivos para revisar su estado y evolución general.			
Funcionalidad			
El sistema debe listar todos los proyectos registrados, mostrando información relevante de cada uno, en una vista clara y ordenada.			
Criterios de aceptación	<div><input type="checkbox"/> Mostrar nombre del proyecto, cliente asociado, estado, plazo estimado y fecha de creación.</div> <div><input type="checkbox"/> El estado debe poder visualizarse con íconos o colores (opcional).</div> <div><input type="checkbox"/> Si no hay proyectos, debe mostrarse un mensaje como “No hay proyectos registrados”.</div>		
Restricciones			
No mostrar campos internos como _id.			

HISTORIA DE USUARIO 14			
Prioridad: Alta			
CÓDIGO DEL REQUERIMIENTO:	RF14	Actor	Freelancer
NOMBRE DEL REQUERIMIENTO	Filtrar proyectos por cliente		
Descripción			
Como Freelancer quiero consultar los proyectos de un cliente específico, para tener visibilidad individual de sus entregables y avances.			
Funcionalidad			
El sistema debe permitir seleccionar un cliente y visualizar únicamente los proyectos asociados a este.			
Criterios de aceptación	<div><input type="checkbox"/> Se debe permitir seleccionar un cliente para aplicar el filtro.</div> <div><input type="checkbox"/> Mostrar los mismos datos que en la HU13.</div> <div><input type="checkbox"/> Si el cliente no tiene proyectos, mostrar un mensaje informativo.</div>		
Restricciones			
El filtrado debe basarse en el campo clienteld.			

HISTORIA DE USUARIO 15			
Prioridad: Alta			
CÓDIGO DEL REQUERIMIENTO:	RF15	Actor	Freelancer
NOMBRE DEL REQUERIMIENTO	Cambiar estado de un proyecto		
Descripción			
Como Freelancer quiero actualizar el estado de un proyecto para reflejar su situación actual (activo, pausado, finalizado o cancelado).			
Funcionalidad			
El sistema debe permitir seleccionar un proyecto y cambiar su estado. No se deben permitir reactivaciones si el estado es finalizado o cancelado.			
Criterios de aceptación	<div><input type="checkbox"/> Permitir cambiar estado a "activo", "pausado", "finalizado" o "cancelado".</div> <div><input type="checkbox"/> No se puede volver a activar un proyecto finalizado o cancelado.</div> <div><input type="checkbox"/> Mostrar confirmación del cambio exitoso.</div>		
Restricciones			
No se deben permitir cambios reversibles para proyectos cerrados (finalizado o cancelado).			


HISTORIA DE USUARIO 16			
Prioridad: Media			
CÓDIGO DEL REQUERIMIENTO:	RF16	Actor	Freelancer
NOMBRE DEL REQUERIMIENTO	Eliminar un proyecto registrado		
Descripción			
Como Freelancer quiero poder eliminar proyectos que no se llevaron a cabo o fueron pruebas, para mantener el sistema limpio y ordenado.			
Funcionalidad			
El sistema debe mostrar una lista de proyectos elegibles para eliminación, validar dependencias como contratos o entregables y requerir confirmación explícita.			
Criterios de aceptación	<ul style="list-style-type: none">Listar solo proyectos que se pueden eliminar.Confirmación previa obligatoria.Si un proyecto tiene contrato o entregables asociados, impedir la eliminación o mostrar advertencia.		
Restricciones			
Proyectos con relaciones activas no pueden ser eliminados automáticamente.			

HISTORIA DE USUARIO 17			
Prioridad: Media			
CÓDIGO DEL REQUERIMIENTO:	RF17	Actor	Usuario (CLI)
NOMBRE DEL REQUERIMIENTO	Salir del submenú de proyectos		
Descripción			
Como usuario quiero poder regresar al menú principal desde el módulo de proyectos, para continuar con otra sección del sistema sin reiniciar la aplicación.			
Funcionalidad			
El submenú de proyectos debe incluir una opción de retorno al menú principal (menuPrincipal()), sin generar errores o salidas inesperadas.			
Criterios de aceptación	<div><input type="checkbox"/> Opción “ Volver” debe estar disponible en el submenú.</div> <div><input type="checkbox"/> Al seleccionar “Volver”, debe redirigir al menú principal correctamente.</div> <div><input type="checkbox"/> El sistema debe continuar operativo.</div>		
Restricciones			
No debe llamarse process.exit() ni provocar reinicio del programa.			

HISTORIA DE USUARIO 18			
Prioridad: Alta			
CÓDIGO DEL REQUERIMIENTO:	RF18	Actor	Freelancer
NOMBRE DEL REQUERIMIENTO	Crear contrato para proyecto existente		
Descripción			
Como Freelancer quiero poder generar un contrato formal para un proyecto existente, incluyendo condiciones, fechas y valor total, para formalizar el acuerdo con el cliente.			
Funcionalidad			
El sistema debe permitir registrar un contrato vinculado a un proyecto que aún no lo tenga. Se deben ingresar todos los campos requeridos, validarlos adecuadamente y asociar el contrato al proyecto correspondiente.			
Criterios de aceptación	<div><input type="checkbox"/> Permitir seleccionar un proyecto que no tenga contrato.</div> <div><input type="checkbox"/> Solicitar condiciones del contrato, fecha de inicio, fecha de fin y valor total.</div> <div><input type="checkbox"/> Validar todos los campos antes de guardar.</div> <div><input type="checkbox"/> Asociar correctamente el contrato al proyecto seleccionado.</div>		
Restricciones			
Cada proyecto puede tener solo un contrato. No debe permitirse múltiples contratos por proyecto.			

HISTORIA DE USUARIO 19			
Prioridad: Media			
CÓDIGO DEL REQUERIMIENTO:	RF19	Actor	Freelancer
NOMBRE DEL REQUERIMIENTO	Consultar contrato vinculado a un proyecto		
Descripción			
Como Freelancer quiero poder visualizar fácilmente el contrato de un proyecto determinado, para revisar los acuerdos y condiciones pactadas.			
Funcionalidad			
El sistema debe permitir seleccionar un proyecto y mostrar su contrato si existe. Si no hay contrato, debe mostrar un mensaje claro al usuario.			
Criterios de aceptación	<div><input type="checkbox"/> Permitir seleccionar un proyecto específico.</div> <div><input type="checkbox"/> Si el proyecto tiene contrato, mostrar condiciones, fechas y valor.</div> <div><input type="checkbox"/> Si no tiene contrato, mostrar mensaje tipo “No se ha generado contrato para este proyecto”.</div>		
Restricciones			
No debe mostrarse información incompleta ni de contratos no asociados correctamente.			

HISTORIA DE USUARIO 20			
Prioridad: Media			
CÓDIGO DEL REQUERIMIENTO:	RF20	Actor	Freelancer
NOMBRE DEL REQUERIMIENTO	Ver todos los contratos registrados		
Descripción			
Como Freelancer quiero tener una vista general de todos los contratos generados en el sistema, para su revisión y control administrativo.			
Funcionalidad			
El sistema debe permitir listar todos los contratos existentes, con datos relevantes para la gestión: proyecto relacionado, cliente, condiciones resumidas, fechas y valor total.			
Criterios de aceptación	<div><input type="checkbox"/> Mostrar proyecto asociado, nombre del cliente, fechas (inicio/fin), condiciones resumidas y valor total.</div> <div><input type="checkbox"/> Si hay muchos registros, permitir desplazamiento paginado o por scroll (CLI-friendly).</div> <div><input type="checkbox"/> Si no hay contratos, mostrar mensaje adecuado.</div>		
Restricciones			
Debe mostrarse solo la información de contratos con asociaciones válidas (a proyectos y clientes existentes).			

HISTORIA DE USUARIO 21			
Prioridad: Media			
CÓDIGO DEL REQUERIMIENTO:	RF21	Actor	Usuario (CLI)
NOMBRE DEL REQUERIMIENTO	Salir del submenú de contratos		
Descripción			
Como usuario quiero poder volver al menú principal desde el módulo de contratos para gestionar otras secciones del sistema.			
Funcionalidad			
El submenú de contratos debe ofrecer una opción de retorno limpio al menú principal, sin reiniciar la aplicación ni ejecutar acciones no deseadas.			
Criterios de aceptación	<div><input type="checkbox"/> Opción “ Volver” debe estar siempre visible.</div> <div><input type="checkbox"/> Al seleccionarla, se debe redirigir correctamente al menuPrincipal().</div> <div><input type="checkbox"/> El flujo debe continuar funcionando correctamente, sin errores ni cierres inesperados.</div>		
Restricciones			
No se debe ejecutar process.exit() ni funciones colaterales durante el retorno.			

HISTORIA DE USUARIO 22			
Prioridad: Alta			
CÓDIGO DEL REQUERIMIENTO:	RF22	Actor	Freelancer
NOMBRE DEL REQUERIMIENTO	Crear entregable asociado a proyecto		
Descripción			
Como Freelancer quiero registrar entregables para un proyecto determinado, incluyendo su descripción, fecha límite y estado inicial, para poder hacer seguimiento detallado.			
Funcionalidad			
El sistema debe permitir crear un entregable para un proyecto activo. Todos los campos deben ser validados y el entregable se debe asociar correctamente al proyecto en la base de datos con estado inicial “pendiente”.			
Criterios de aceptación	<div><input type="checkbox"/> Permitir seleccionar un proyecto activo.</div> <div><input type="checkbox"/> Solicitar nombre, descripción, fecha límite y asignar estado "pendiente" automáticamente.</div> <div><input type="checkbox"/> Validar que la fecha límite sea futura.</div> <div><input type="checkbox"/> Asociar el entregable correctamente al proyecto.</div>		
Restricciones			
No se deben crear entregables para proyectos en estado finalizado o cancelado.			

HISTORIA DE USUARIO 23			
Prioridad: Media			
CÓDIGO DEL REQUERIMIENTO:	RF23	Actor	Freelancer
NOMBRE DEL REQUERIMIENTO	Ver entregables de un proyecto específico		
Descripción			
Como Freelancer quiero visualizar todos los entregables asociados a un proyecto para saber su estado y planificar entregas.			
Funcionalidad			
El sistema debe permitir seleccionar un proyecto y listar todos sus entregables mostrando la información necesaria para el seguimiento.			
Criterios de aceptación	<div><input type="checkbox"/> Mostrar tabla con nombre, fecha límite, estado y fecha de creación.</div> <div><input type="checkbox"/> Mostrar mensaje informativo si el proyecto no tiene entregables.</div>		
Restricciones			
No mostrar entregables de otros proyectos en esta vista.			

HISTORIA DE USUARIO 24			
Prioridad: Alta			
CÓDIGO DEL REQUERIMIENTO:	RF24	Actor	Freelancer
NOMBRE DEL REQUERIMIENTO	Actualizar estado de un entregable		
Descripción			
Como Freelancer quiero poder marcar un entregable como entregado, aprobado o rechazado para reflejar su avance real.			
Funcionalidad			
El sistema debe permitir cambiar el estado de un entregable existente a uno de los cuatro posibles estados. Si el estado es "rechazado", debe solicitar una razón obligatoria. No se permite modificar un entregable aprobado.			
Criterios de aceptación	<div><input type="checkbox"/> Mostrar entregables de un proyecto seleccionado.</div> <div><input type="checkbox"/> Permitir cambiar estado a: pendiente, entregado, aprobado, rechazado.</div> <div><input type="checkbox"/> Al marcar como rechazado, se debe solicitar una razón.</div> <div><input type="checkbox"/> No permitir cambiar estado si ya está como aprobado.</div>		
Restricciones			
El estado "aprobado" es definitivo. No debe revertirse.			

HISTORIA DE USUARIO 25			
Prioridad: Media			
CÓDIGO DEL REQUERIMIENTO:	RF25	Actor	Freelancer
NOMBRE DEL REQUERIMIENTO	Eliminar entregable de un proyecto		
Descripción			
Como Freelancer quiero eliminar un entregable que fue creado por error o cancelado para mantener limpio el sistema.			
Funcionalidad			
El sistema debe permitir eliminar entregables pendientes o rechazados, con confirmación previa. La operación debe ejecutarse dentro de una transacción para garantizar integridad si hay dependencias.			
Criterios de aceptación	<div><input type="checkbox"/> Mostrar lista de entregables por proyecto.</div> <div><input type="checkbox"/> Confirmar con el usuario antes de eliminar.</div> <div><input type="checkbox"/> No eliminar entregables con estado "aprobado".</div>		
Restricciones			
Debe implementarse lógica de protección para entregables ya aprobados.			

HISTORIA DE USUARIO 26			
Prioridad: Media			
CÓDIGO DEL REQUERIMIENTO:	RF26	Actor	Usuario (CLI)
NOMBRE DEL REQUERIMIENTO	Salir del submenú de entregables		
Descripción			
Como usuario quiero volver al menú principal desde el módulo de entregables para navegar a otras áreas del sistema.			
Funcionalidad			
Cada submenú de entregables debe incluir una opción para retornar al menuPrincipal() sin reiniciar la app ni generar errores.			
Criterios de aceptación	<div><input type="checkbox"/> Mostrar opción “🔙 Volver” en el submenú de entregables.</div> <div><input type="checkbox"/> Al elegirla, retornar correctamente al menú principal.</div> <div><input type="checkbox"/> No ejecutar acciones adicionales ni cerrar el sistema.</div>		
Restricciones			
No debe llamarse process.exit() dentro de este flujo de retorno.			

HISTORIA DE USUARIO 27			
Prioridad: Alta			
CÓDIGO DEL REQUERIMIENTO:	RF27	Actor	Freelancer
NOMBRE DEL REQUERIMIENTO	Registrar ingreso asociado a un proyecto		
Descripción			
Como Freelancer quiero registrar ingresos relacionados con proyectos, como anticipos o pagos completos, para llevar el control de mis entradas de dinero.			
Funcionalidad			
El sistema debe permitir seleccionar un proyecto válido e ingresar información del ingreso (monto, descripción, fecha). El tipo debe asignarse automáticamente como "ingreso".			
Criterios de aceptación	<ul style="list-style-type: none">• Seleccionar un proyecto existente y válido.• Ingresar monto positivo, descripción mínima de 5 caracteres y fecha no futura.• Tipo de operación: "ingreso" (asignado automáticamente).• El ingreso debe vincularse correctamente al proyecto.		
Restricciones			
El ingreso debe estar vinculado a un proyecto con contrato (si se aplica lógica de contrato activo).			

HISTORIA DE USUARIO 28			
Prioridad: Alta			
CÓDIGO DEL REQUERIMIENTO:	RF28	Actor	Freelancer
NOMBRE DEL REQUERIMIENTO	Registrar egreso asociado a un proyecto		
Descripción			
Como Freelancer quiero registrar egresos relacionados a un proyecto (gastos operativos, herramientas, etc.) para controlar mis salidas de dinero.			
Funcionalidad			
El sistema debe registrar egresos con los mismos campos que un ingreso, pero asignando el tipo "egreso". Puede validarse que el egreso no supere el saldo disponible del proyecto.			
Criterios de aceptación	<div><input type="checkbox"/> Seleccionar un proyecto válido.</div> <div><input type="checkbox"/> Ingresar monto, descripción, fecha válida.</div> <div><input type="checkbox"/> Tipo de operación debe ser "egreso".</div> <div><input type="checkbox"/> Validar que no supere los ingresos totales del proyecto (opcional si hay control de saldo).</div>		
Restricciones			
Si hay control de saldo, no deben registrarse egresos que dejen el balance en negativo.			

HISTORIA DE USUARIO 29			
Prioridad: Alta			
CÓDIGO DEL REQUERIMIENTO:	RF29	Actor	Freelancer
NOMBRE DEL REQUERIMIENTO	Consultar balance financiero por cliente		
Descripción			
Como Freelancer quiero ver el balance general de ingresos y egresos para cada cliente, para evaluar rentabilidad.			
Funcionalidad			
El sistema debe permitir seleccionar un cliente y mostrar el resumen financiero consolidado de todos sus proyectos: ingresos, egresos y balance neto.			
Criterios de aceptación	<div><input type="checkbox"/> Seleccionar cliente existente.</div> <div><input type="checkbox"/> Mostrar: ingresos totales, egresos totales, balance neto.</div> <div><input type="checkbox"/> Incluir todos los proyectos del cliente en el cálculo.</div>		
Restricciones			
El balance debe incluir únicamente proyectos con movimientos financieros registrados.			

HISTORIA DE USUARIO 31			
Prioridad: Media			
CÓDIGO DEL REQUERIMIENTO:	RF31	Actor	Freelancer
NOMBRE DEL REQUERIMIENTO	Listar todas las operaciones financieras		
Descripción			
Como Freelancer quiero ver todos los ingresos y egresos registrados en el sistema para tener visibilidad detallada.			
Funcionalidad			
El sistema debe listar todas las operaciones financieras registradas, ordenadas cronológicamente, indicando tipo, monto, fecha, proyecto y descripción.			
Criterios de aceptación	<div><input type="checkbox"/> Mostrar operaciones con: fecha, tipo (ingreso / egreso), descripción, monto, proyecto.</div> <div><input type="checkbox"/> Mostrar en orden cronológico descendente.</div> <div><input type="checkbox"/> Si no hay operaciones, indicar mensaje adecuado.</div>		
Restricciones			
Debe garantizarse que las operaciones estén correctamente asociadas a sus proyectos.			

HISTORIA DE USUARIO 32			
Prioridad: Media			
CÓDIGO DEL REQUERIMIENTO:	RF32	Actor	Usuario (CLI)
NOMBRE DEL REQUERIMIENTO	Salir del módulo financiero		
Descripción			
Como usuario quiero regresar al menú principal desde la gestión financiera para navegar otras funciones del sistema.			
Funcionalidad			
El módulo financiero debe incluir la opción “🔙 Volver”, que retorne al menuPrincipal() sin cerrar el sistema ni afectar el estado general.			
Criterios de aceptación	<div><input type="checkbox"/> Opción “🔙 Volver” siempre visible.</div> <div><input type="checkbox"/> Al seleccionarla, redirigir correctamente al menú principal.</div> <div><input type="checkbox"/> No ejecutar funciones inesperadas ni salir del sistema.</div>		
Restricciones			
No debe ejecutarse process.exit() dentro de este flujo.			

HISTORIA DE USUARIO 33			
Prioridad: Alta			
CÓDIGO DEL REQUERIMIENTO:	RF33	Actor	Sistema
NOMBRE DEL REQUERIMIENTO	Validar que todos los campos requeridos sean correctos		
Descripción			
Como desarrollador quiero que el sistema valide todos los campos de entrada en formularios CLI para evitar errores y datos corruptos en la base de datos.			
Funcionalidad			
Antes de almacenar cualquier dato, el sistema debe validar que todos los campos requeridos estén presentes, tengan el tipo de dato correcto y cumplan con las reglas mínimas de validación (longitud, formato, rango, etc.).			
Criterios de aceptación	<div><input type="checkbox"/> Todos los campos requeridos deben validarse (strings, números, fechas).</div> <div><input type="checkbox"/> Impedir persistencia de campos vacíos o con valores nulos.</div> <div><input type="checkbox"/> Mostrar mensajes claros en consola cuando se detecte un error de validación.</div>		
Restricciones			
Las validaciones deben ejecutarse antes de cualquier operación de inserción o actualización.			

HISTORIA DE USUARIO 35			
Prioridad: Alta			
CÓDIGO DEL REQUERIMIENTO:	RF35	Actor	Sistema
NOMBRE DEL REQUERIMIENTO	Validar consistencia entre datos relacionados		
Descripción			
Como desarrollador quiero que el sistema valide que las entidades relacionadas existan y se correspondan antes de realizar operaciones (por ejemplo, crear una propuesta con cliente válido o un contrato con proyecto existente).			
Funcionalidad			
Antes de guardar relaciones entre colecciones, el sistema debe verificar que los IDs referenciados existen y son válidos, evitando referencias huérfanas o inválidas.			
Criterios de aceptación	<div><input type="checkbox"/> Validar la existencia de clienteld antes de asociar en propuestas, proyectos o contratos.</div> <div><input type="checkbox"/> Validar que proyectold exista antes de registrar entregables, ingresos, egresos o contratos.</div> <div><input type="checkbox"/> Cancelar operaciones si las relaciones no existen, con mensaje claro al usuario.</div>		
Restricciones			
No debe permitirse el uso de IDs inexistentes ni incorrectos en relaciones cruzadas.			

HISTORIA DE USUARIO 36			
Prioridad: Alta			
CÓDIGO DEL REQUERIMIENTO:	RF36	Actor	Usuario (CLI)
NOMBRE DEL REQUERIMIENTO	Mostrar errores claros al usuario final		
Descripción			
Como usuario quiero recibir mensajes comprensibles cuando ocurre un error, para entender qué sucedió y cómo solucionarlo.			
Funcionalidad			
Cuando el sistema detecte un error (validación, relación, operación fallida), debe comunicarlo mediante mensajes amigables y claros. No deben mostrarse errores del sistema (como stack traces) al usuario final.			
Criterios de aceptación	<div><input type="checkbox"/> Todos los errores deben estar redactados de forma comprensible para el usuario.</div> <div><input type="checkbox"/> No mostrar errores técnicos como excepciones no controladas.</div> <div><input type="checkbox"/> Incluir sugerencias o invitaciones a repetir la operación o volver al menú principal.</div> <div><input type="checkbox"/> Los mensajes de éxito pueden mostrarse en color verde si se usa una librería como chalk.</div>		
Restricciones			
Ningún error sin manejo debe alcanzar la vista del usuario (uso obligatorio de try/catch y control por capas).			

5. METODOLOGÍA

Se adopta la metodología ágil **SCRUM**, adaptada a una ejecución individual:

- **Scrum Master:** Isaac Medina
- **Product Owner:** Isaac Medina
- **Developer:** Isaac Medina
- Se realizará un único sprint de 7 días que incluye las ceremonias principales: Sprint Planning, Daily Stand-ups (individuales), Sprint Review y Retrospective. El seguimiento del proyecto se realiza mediante un tablero de GitHub Projects, documentando tareas y su progreso.

6. EVIDENCIA DE PLANTEAMIENTO DE PLATAFORMA DE TRABAJO

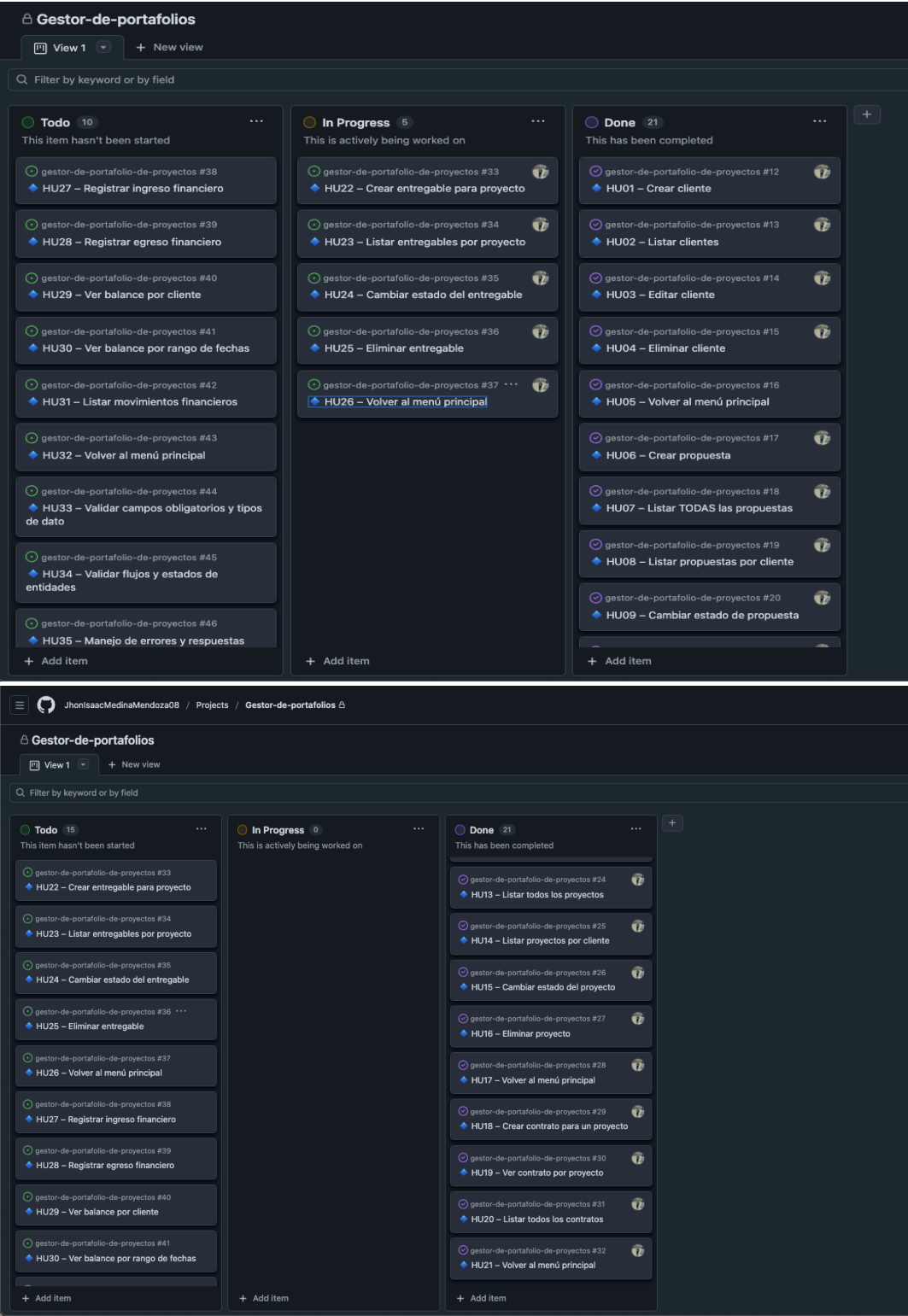
Repositorio GitHub:

<https://github.com/JhonIsaacMedinaMendoza08/gestor-de-portafolio-de-proyectos>

Videos de presentación Proyecto

<https://youtu.be/moJxnCaxFZE>

Capturas del tablero Scrum:



JhonIsaacMedinaMendoza08 / Projects / Gestor-de-portafolios

Gestor-de-portafolios

View 1

+ New view

Filter by keyword or by field

Todo 4

This item hasn't been started

gestor-de-portafolio-de-proyectos #44

HU33 - Validar campos obligatorios y tipos de dato

gestor-de-portafolio-de-proyectos #45

HU34 - Validar flujos y estados de entidades

gestor-de-portafolio-de-proyectos #46

HU35 - Manejo de errores y respuestas amigables

gestor-de-portafolio-de-proyectos #47

HU36 - Interfaz de usuario estética y clara en consola

+ Add item

In Progress 5

This is actively being worked on

gestor-de-portafolio-de-proyectos #38

HU27 - Registrar Ingreso financiero

gestor-de-portafolio-de-proyectos #39

HU28 - Registrar egreso financiero

gestor-de-portafolio-de-proyectos #40

HU29 - Ver balance por cliente

gestor-de-portafolio-de-proyectos #41

HU30 - Ver balance por rango de fechas

gestor-de-portafolio-de-proyectos #42

HU31 - Listar movimientos financieros

+ Add item

Done 27

This has been completed

gestor-de-portafolio-de-proyectos #43

HU32 - Volver al menú principal

gestor-de-portafolio-de-proyectos #12

HU01 - Crear cliente

gestor-de-portafolio-de-proyectos #13

HU02 - Listar clientes

gestor-de-portafolio-de-proyectos #14

HU03 - Editar cliente

gestor-de-portafolio-de-proyectos #15

HU04 - Eliminar cliente

gestor-de-portafolio-de-proyectos #16

HU05 - Volver al menú principal

gestor-de-portafolio-de-proyectos #17

HU06 - Crear propuesta

gestor-de-portafolio-de-proyectos #18

HU07 - Listar TODAS las propuestas

gestor-de-portafolio-de-proyectos #19

HU08 - Listar propuestas por cliente

+ Add item

JhonIsaacMedinaMendoza08 / Projects / Gestor-de-portafolios

Gestor-de-portafolios

View 1

+ New view

Filter by keyword or by field

Todo 0

This item hasn't been started

+ Add item

In Progress 0

This is actively being worked on

+ Add item

Done 34

This has been completed

HU26 - Volver al menú principal

gestor-de-portafolio-de-proyectos #35

HU24 - Cambiar estado del entregable

gestor-de-portafolio-de-proyectos #38

HU27 - Registrar ingreso financiero

gestor-de-portafolio-de-proyectos #39

HU28 - Registrar egreso financiero

gestor-de-portafolio-de-proyectos #40

HU29 - Ver balance por cliente

gestor-de-portafolio-de-proyectos #42

HU31 - Listar movimientos financieros

gestor-de-portafolio-de-proyectos #44

HU33 - Validar campos obligatorios y tipos de dato

gestor-de-portafolio-de-proyectos #46

HU35 - Manejo de errores y respuestas amigables

gestor-de-portafolio-de-proyectos #47

HU36 - Interfaz de usuario estética y clara en consola

+ Add item

<https://github.com/JhonIsaacMedinaMendoza08/gestor-de-portafolio-de-proyectos/issues/46>

Tecnologías utilizadas:

- Node.js
- MongoDB (driver oficial)
- Inquirer, Chalk, Dotenv, Dayjs, ajv, ajv-Formats
- Principios SOLID
- Patrones de diseño: Repository y Factory

Resultado funcional:

Aplicación CLI funcional que permite al freelancer gestionar su portafolio completo desde consola.

7. CONCLUSIONES

El desarrollo del sistema “**Gestor de Portafolio de Proyectos Freelance**” representa una **aplicación funcional orientada a casos reales**, ya que responde directamente a una problemática común que enfrentan los freelancers: la dificultad de organizar y hacer seguimiento eficiente a múltiples clientes, propuestas, proyectos, entregables y movimientos financieros. Esta herramienta, construida como una aplicación de línea de comandos (CLI), ofrece una interfaz centralizada que permite gestionar de manera profesional todo el ciclo de trabajo del freelancer, desde el contacto inicial con el cliente hasta el control del balance económico.

El proyecto fue implementado siguiendo un **proceso profesional de desarrollo**, basado en buenas prácticas de ingeniería de software. Durante su construcción se aplicaron los **principios SOLID** para garantizar un código limpio, mantenible y escalable. Asimismo, se implementaron **patrones de diseño como Repository y Factory**, además de un enfoque estructurado en módulos, validaciones exhaustivas de datos, relaciones entre entidades y un sistema de manejo de errores claro y amigable para el usuario final.

Para la gestión del proyecto se utilizó **la metodología ágil SCRUM adaptada a un entorno individual**. Este enfoque no solo permitió mantener una organización clara del trabajo, sino que también simuló un entorno de desarrollo profesional, disciplinado y enfocado en la entrega incremental de valor.

Finalmente, gracias a su arquitectura modular y a la separación por responsabilidades, la aplicación no solo es útil para trabajadores independientes,

sino que tiene una **aplicabilidad extendida**. Puede adaptarse fácilmente a contextos más amplios como pequeñas agencias, equipos de desarrollo técnico o startups que necesiten una solución ligera y potente para gestionar sus portafolios desde consola. Esto convierte al proyecto no solo en una práctica académica, sino en una base sólida para sistemas de gestión reales en el mundo profesional.

Conclusiones de la Reunión de Retrospectiva del Sprint

1. Lo que se hizo bien:

- Se documentaron todos los requerimientos antes de desarrollar.
- Se logró cumplir con todas las historias planificadas del Sprint.
- El sistema fue probado constantemente durante el desarrollo.

2. Lo que se puede mejorar:

- Usar herramientas de testing automatizado en futuras versiones.
- Profundizar en seguridad (ej. autenticación) para producción.
- Posiblemente extender el sistema a interfaz web en el siguiente módulo.