

# GESTOR DE PORTAFOLIO PARA FREELANCERS

Mateo Paternina Mercado

GRUPO: INDIVIDUAL

DOCENTES: EQUIPO CAMPUSLANDS

CAMPUSLANDS

SALÓN: U2

RUTA: NODE.JS

FLORIDABLANCA - 2025

## 1. TÍTULO DEL PROYECTO

Gestor de portafolio para freelancers

## 2. SITUACIÓN PROBLEMA

Los freelancers enfrentan el reto de mantener organizada toda su operación profesional: clientes, propuestas, proyectos, contratos, entregables y finanzas. No contar con una herramienta integral puede llevar a pérdida de oportunidades, errores administrativos y falta de control financiero. Este proyecto surge para crear una herramienta CLI que permita al freelancer centralizar toda esta información y operar de manera más eficiente, sin depender de múltiples sistemas dispersos.

## 3. LEVANTAMIENTO DE REQUERIMIENTOS

Requerimientos Funcionales:

- RF01: CRUD de clientes
- RF02: CRUD de propuestas
- RF03: Generar proyecto al aceptar propuesta
- RF04: CRUD de proyectos
- RF05: Asignación y control de entregables
- RF06: Gestión de contratos
- RF07: Registro de ingresos y egresos
- RF08: Uso de transacciones para entregables y finanzas

Requerimientos No Funcionales:

- RNF01: La aplicación debe ejecutarse por línea de comandos
- RNF02: Uso de MongoDB con el driver oficial
- RNF03: Validación manual de campos
- RNF04: Arquitectura modular, principios SOLID y patrones de diseño

## 4. HISTORIAS DE USUARIO CON CRITERIOS DE ACEPTACIÓN

Historia de usuario:

Prioridad: Alta

Código del requerimiento: RF03

Actor: Freelancer (Administrador)

Nombre: Generación automática de proyectos

Descripción: Como freelancer quiero que al aceptar una propuesta, se genere automáticamente un proyecto asociado, para agilizar el flujo de trabajo.

Funcionalidad: Detectar propuestas aceptadas y convertirlas en un nuevo proyecto con los datos básicos ya cargados.

Criterios de aceptación:

1. El sistema debe crear un nuevo proyecto al aceptar una propuesta.
2. El sistema debe asociar automáticamente al cliente y propuesta al proyecto.
3. El sistema debe marcar la propuesta como usada para evitar duplicidad.

Restricciones: Solo propuestas en estado 'aceptada' pueden generar un proyecto.

## **5. METODOLOGÍA**

Se utiliza la metodología ágil SCRUM para la planeación y seguimiento del desarrollo del proyecto.

Roles:

- Scrum Master: Mateo Paternina
- Product Owner: Mateo Paternina
- Developer: Mateo Paternina

Cada sprint tiene una duración de 1 a 2 días, y las tareas son administradas desde GitHub Projects.

## **6. EVIDENCIA DE PLATAFORMA DE TRABAJO**

- Repositorio del proyecto: [Privado en GitHub, acceso otorgado al docente]
- Videos de reuniones: Grabaciones individuales de sprint planning y demos
- Tablero SCRUM: GitHub Projects con tareas, responsables, prioridades y tiempos
- Documentación del producto: README completo, principios SOLID y patrones aplicados

## **7. CONCLUSIONES**

- La CLI provee una solución práctica y profesional para freelancers organizados.
- MongoDB, junto con validaciones y transacciones, asegura la integridad de datos.
- El enfoque modular y la metodología ágil facilitaron el cumplimiento del alcance.
- En la retrospectiva se destacó la importancia de un plan de trabajo claro incluso al trabajar de manera individual.