

Console-Fiverr – Gestión de Portafolio para Freelancers vía CLI

NOMBRES DE LOS ESTUDIANTES

Daniel Felipe Florez Cubides

GRUPO ASIGNADO

Grupo 1 - Backend [Node.js](#)

DOCENTES INVOLUCRADOS

Juan Mariño

**CAMPUSLANDS
SALÓN ARTEMIS
RUTA NODE.JS
FLORIDABLANCA
2025**

1. SITUACIÓN PROBLEMA

Los freelancers suelen llevar un control disperso y poco sistematizado de sus proyectos, clientes y finanzas, recurriendo a hojas de cálculo o notas personales. Esto dificulta el seguimiento profesional, especialmente cuando se gestionan múltiples propuestas, contratos y entregables simultáneamente.

Console-Fiverr nace como una solución de línea de comandos que permite unificar y organizar todos los elementos clave del portafolio freelance con validaciones, operaciones financieras seguras y control total del ciclo de vida del proyecto.

2. LEVANTAMIENTO DE REQUERIMIENTOS

El levantamiento de requerimientos se hizo considerando el flujo completo de un profesional freelance, desde el contacto con el cliente hasta el cierre financiero del proyecto.

3. REQUERIMIENTOS

3.1 Requerimientos funcionales

- CRUD de clientes, proyectos, contratos y entregables
- Creación automática de proyectos a partir de propuestas aceptadas
- Registro de entregables con fechas límite y estados
- Registro de ingresos y egresos asociados a cada proyecto
- Cálculo de balances financieros por cliente y proyecto
- Exportación de movimientos en CSV y JSON
- Menús interactivos vía CLI usando inquirer
- Persistencia en MongoDB
- Uso de transacciones en operaciones críticas

3.2 Requerimientos no funcionales

- Sistema usable completamente desde la consola
- Validaciones a nivel de modelo (tipos, fechas, formatos)
- Persistencia en MongoDB con driver oficial
- Código estructurado por SOLID y patrones Repository + Service
- Proyecto organizado por módulos reutilizables y escalables

4. HISTORIAS DE USUARIO CON CRITERIOS DE ACEPTACIÓN

HISTORIA DE USUARIO			
Prioridad: Alta			
CÓDIGO DEL REQUERIMIENTO:	HU-001	Actor	Usuario CLI (Freelancer)
NOMBRE DEL REQUERIMIENTO	Gestión de clientes		
Descripción			
Como freelancer quiero registrar y gestionar clientes para asociarlos a propuestas y proyectos.			
Funcionalidad			
Permite registrar, actualizar, eliminar y listar clientes con sus datos.			
Criterios de aceptación	1 El sistema debe permitir registrar clientes con validaciones. 2 El sistema debe listar clientes en tabla con formato ordenado. 3 El sistema debe permitir seleccionar clientes sin usar IDs.		
Restricciones			
No se deben eliminar clientes asociados a propuestas o proyectos activos.			

HISTORIA DE USUARIO			
Prioridad: Alta			
CÓDIGO DEL REQUERIMIENTO:	HU-002	Actor	Usuario CLI (Freelancer)
NOMBRE DEL REQUERIMIENTO	Gestión de Propuestas		
Descripción			
Como freelancer quiero crear propuestas para mis clientes con precio, plazo y estado, para que puedan ser aceptadas o rechazadas.			
Funcionalidad			
El sistema permite crear propuestas y, si son aceptadas, genera automáticamente un nuevo proyecto.			
Criterios de aceptación	1. El sistema debe permitir registrar propuestas con campos obligatorios. 2 Al aceptar una propuesta, se debe crear automáticamente un proyecto. 3 El estado debe cambiar entre "pendiente", "aceptada" y "rechazada".		
Restricciones			
Solo se puede tener una propuesta activa por cliente al mismo tiempo.			

HISTORIA DE USUARIO			
Prioridad: Alta			
CÓDIGO DEL REQUERIMIENTO:	HU-003	Actor	Usuario CLI (Freelancer)
NOMBRE DEL REQUERIMIENTO	Gestión de Proyectos		
Descripción			
Como freelancer quiero gestionar proyectos y registrar avances para tener control sobre su estado y progreso.			
Funcionalidad			
Permite listar, actualizar y cambiar el estado del proyecto, así como registrar avances en su desarrollo.			
Criterios de aceptación	1. El sistema debe mostrar los proyectos y su estado actual. 2 Se debe permitir cambiar el estado entre activo, pausado, finalizado y cancelado. 3 Se debe permitir registrar avances con fecha y detalle.		
Restricciones			
No se deben modificar proyectos finalizados o cancelados.			

HISTORIA DE USUARIO			
Prioridad: Media			
CÓDIGO DEL REQUERIMIENTO:	HU-004	Actor	Usuario CLI (Freelancer)
NOMBRE DEL REQUERIMIENTO	Gestión de Contratos		
Descripción			
Como freelancer quiero asociar contratos a los proyectos para dejar claras las condiciones de trabajo.			
Funcionalidad			
El sistema permite registrar un contrato con condiciones, fechas y valor, y asociarlo a un proyecto específico.			
Criterios de aceptación	1. El sistema debe validar que las fechas y valores sean coherentes. 2 Se debe listar contratos existentes. 3 Se debe mostrar el detalle del contrato seleccionado.		
Restricciones			
Un proyecto solo puede tener un contrato activo.			

HISTORIA DE USUARIO			
Prioridad: Alta			
CÓDIGO DEL REQUERIMIENTO:	HU-006	Actor	Usuario CLI (Freelancer)
NOMBRE DEL REQUERIMIENTO	Gestión de Finanzas con transacción		
Descripción			
Como freelancer quiero registrar ingresos y egresos asociados a proyectos para tener control financiero.			
Funcionalidad			
Permite ingresar movimientos financieros con tipo, monto, fecha, descripción y los guarda dentro de una transacción MongoDB.			
Criterios de aceptación	1. El sistema debe validar monto y tipo del movimiento. 2 El registro debe realizarse dentro de una transacción. 3 Se debe poder listar movimientos por proyecto y por rango de fechas.		
Restricciones			
No se pueden registrar movimientos con monto ≤ 0.			

HISTORIA DE USUARIO			
Prioridad: Media			
CÓDIGO DEL REQUERIMIENTO:	HU-007	Actor	Usuario CLI (Freelancer)
NOMBRE DEL REQUERIMIENTO	Exportación de movimientos		
Descripción			
Como freelancer quiero exportar los movimientos financieros en CSV o JSON para tener un respaldo.			
Funcionalidad			
Se puede elegir proyecto y formato, y se guarda el archivo en una carpeta local.			
Criterios de aceptación	1. El sistema debe permitir seleccionar el formato entre JSON y CSV. 2 El archivo se debe guardar en /exports. 3 Debe confirmarse con un mensaje en consola.		
Restricciones			
Solo se pueden exportar movimientos de proyectos existentes.			

5. METODOLOGÍA

Se utilizó la metodología SCRUM para el desarrollo ágil del proyecto. El equipo fue compuesto por un solo desarrollador cumpliendo todos los roles clave:

Rol	Responsable
Scrum Master	Daniel Felipe Florez Cubides
Product Owner	Daniel Felipe Florez Cubides
Developer	Daniel Felipe Florez Cubides

El desarrollo se dividió en sprints por módulos, con una duración aproximada de 8 horas cada uno.

6. EVIDENCIA DE PLATAFORMA DE TRABAJO

- **Repositorio del proyecto:**
<https://github.com/DanielFelipeFlorezCubides/Console-Fiverr/tree/main>
 - **Herramienta de seguimiento SCRUM:**
GitHub Projects
Se documentaron tareas por sprint y se relacionaron a issues y commits.
 - **Commits y ramas:**
Se aplicó la convención `feat:`, `fix:`, `docs:` en todos los commits
Ejemplo: `feat:` agregar CLI para contratos
 - **Tecnologías usadas:**
 - Node.js + MongoDB (driver oficial)
 - Inquirer, Chalk, dotenv, json2csv
 - Patrón Service + Repository
 - Arquitectura modular basada en CLI y SOLID
-

7. CONCLUSIONES

- El sistema desarrollado cumple con todos los requerimientos funcionales y técnicos planteados.
- La implementación en CLI demostró ser potente y flexible para entornos sin interfaces gráficas.
- Las transacciones y validaciones aseguran integridad de datos en áreas críticas.
- El enfoque SCRUM permitió una planificación y avance estructurado del proyecto.
- La herramienta puede evolucionar a interfaz gráfica o API REST fácilmente.