

Documento de Requerimientos de Software - LexPro

PROPIEDAD EXCLUSIVA DE LEXPRO

Febrero 2026

Tabla de Contenidos

Documento de Requerimientos de Software - LexPro	1
Documento de Requerimientos de Software - LexPro	1
1. Introducción	
2. Requerimientos Funcionales	2
3. Requerimientos No Funcionales	4

Documento de Requerimientos de Software - LexPro

Fecha: Febrero de 2026

Versión: 1.0

1. Introducción

El presente documento detalla los requerimientos funcionales y no funcionales para el sistema de gestión "LexPro". El objetivo principal del sistema es centralizar la operación del despacho, eliminar la reactividad ante los clientes y proporcionar visibilidad completa sobre los plazos y estados de los asuntos legales.

2. Requerimientos Funcionales

Los requerimientos funcionales describen lo que el sistema debe hacer. Están organizados por módulos clave.

2.1. Módulo de Autenticación y Seguridad

- ✓ **RF-001 (Inicio de Sesión):** El sistema debe permitir a los usuarios autenticarse mediante correo electrónico y contraseña segura.
- ✓ **RF-002 (Gestión de Roles):** El sistema debe distinguir entre dos roles principales:
 - ✓ **Administrador (Socios):** Acceso total a todos los módulos, configuración y visión global.
 - ✓ **Abogado (Colaborador):** Acceso limitado a sus proyectos asignados, carga de horas y pendientes.
- ✓ **RF-003 (Recuperación de Contraseña):** Mecanismo para restablecer la contraseña vía correo electrónico.

2.2. Módulo de Gestión de Proyectos y Tareas (Core)

- ✓ **RF-004 (Creación de Proyectos):** Los administradores deben poder crear proyectos asociados a un cliente.
- ✓ **RF-005 (Categorización):** Cada proyecto debe clasificarse según su categoría por un color:

- ✓ **RF-006 (Datos de Litigio):** Para proyectos de tipo "Litigio", el sistema debe permitir capturar campos específicos como:
 - ✓ Juzgado
 - ✓ Número de Expediente
- ✓ **RF-007 (Asignación de Tareas):** Permitir asignar tareas específicas a uno o varios abogados responsables.
- ✓ **RF-008 (Gestión de Plazos):** Cada tarea debe tener una fecha de vencimiento obligatoria para activar el sistema de alertas.
- ✓ **RF-009 (Prioridades):** Clasificación de tareas por nivel de urgencia: Baja, Media, Alta y Crítica.

2.3. Sistema de Notificaciones Inteligentes

- ✓ **RF-010 (Alertas por Vencimiento):** Notificaciones automáticas cuando se acerque una fecha límite (ej. 5 días antes, 2 días antes, mismo día). Según lo configure el admin para cada caso particular.
- ✓ **RF-011 (Canales de Envío):** Las notificaciones críticas deben enviarse tanto por correo electrónico como por WhatsApp (mediante integración API).

2.4. Dashboard y Visualización

- ✓ **RF-012 (Vista de Abogado):** Un panel principal para colaboradores que muestre "Mis Pendientes" ordenados por fecha de vencimiento y demás filtros.
- ✓ **RF-013 (Dashboard Gerencial):** Vista para socios que muestre la carga de trabajo global y alertas críticas de todos los proyectos.
- ✓ **RF-014 (Gráficos de Distribución):** Visualización gráfica (dona/barras) de la distribución de asuntos por categoría.

2.5. Gestión de Clientes (Mini CRM)

- ✓ **RF-015 (Directorio de Clientes):** Registro de información básica del cliente: Nombre Fiscal, Contacto Principal, Correo Electrónico, Teléfono, dirección, etc.

2.6. Registro de Tiempos (Time Tracking)

- ✓ **RF-016 (Carga de Horas):** Los abogados deben poder registrar el tiempo invertido en cada tarea.
 - ✓ **RF-017 (Facturabilidad):** Opción para marcar las horas como "Facturables" o "No Facturables".
-

3. Requerimientos No Funcionales

Los requerimientos no funcionales definen cómo debe comportarse el sistema (calidad, rendimiento, restricciones).

3.1. Infraestructura y Despliegue

- ✓ **RNF-001 (Hosting):** El sistema será autohospedado en un VPS (Servidor Privado Virtual) de DigitalOcean.
- ✓ **RNF-002 (Base de Datos):** Se utilizará **MariaDB** como motor de base de datos relacional.
- ✓ **RNF-003 (Separación de Servicios):** La arquitectura debe separar el servidor de aplicación (Droplet A) del servidor de base de datos (Droplet B o servicio gestionado) para facilitar escalabilidad.

3.2. Arquitectura y Escalabilidad

- ✓ **RNF-010 (Diseño Multi-tenant):** La base de datos y la lógica del backend deben diseñarse desde el inicio para soportar múltiples organizaciones (tenants) de forma aislada.
 - ✓ Todas las tablas principales (usuarios, proyectos, tareas, clientes) deben incluir un campo `organization_id`.
 - ✓ Esto permitirá escalar la solución a otros despachos en el futuro sin necesidad de reescribir el código base.

3.3. Rendimiento y Disponibilidad

- ✓ **RNF-004 (Tecnología Frontend):** Interfaz desarrollada en **Next.js** para garantizar tiempos de carga rápidos y renderizado optimizado.
- ✓ **RNF-005 (Tecnología Backend):** API desarrollada en **NestJS** para asegurar robustez y mantenibilidad.
- ✓ **RNF-006 (Disponibilidad):** El sistema debe estar disponible 24/7

3.4. Usabilidad e Interfaz

- ✓ **RNF-007 (Diseño Responsive):** La aplicación debe ser completamente funcional en dispositivos móviles (smartphones/tablets) y navegadores de escritorio.
 - ✓ **RNF-008 (Facilidad de Uso):** Interfaz intuitiva que no requiera conocimientos técnicos avanzados; orientada a la lectura rápida de pendientes (pensando en usuarios con poco tiempo).
-