**Requisitos Mínimos**

Herramientas del desarrollo del sistema con sus lineamientos:  
  
Laravel: El laravel es un software que te apoya brindando lo que es la estructura de php, y lo necesario para el desarrollo de un sistema web

PHP: Es un lenguaje de desarrollo para el sistema

Workbench: Es un software que se encarga de crear la base de datos para poder configurar el sistema con sus respectivas funciones

Seguridad: JWT, roles básicos, HTTPS:

Logs: Errores y acciones criticas:

Base de datos: Migraciones documentación ERD:

Git: Ramas y PR básicas:

Despliegue: DEV/QA/PROD:

Documentación: Swaggger/OpenAPI:

Observabilidad: Logs básicos:

Gobernanza: Propiedad intelectual protección de datos :

**Herramientas del desarrollo del sistemas completos**

**LDAP + OAuth2 + SSO, rate limiting, pentesting:**Logs estructurados, analítica, ELK Stack:

Auditoría BD, triggers, optimización picos concurrencia:

Flujo profesional feature/release/hotfix, Code Review obligatorio:

Carga, estrés, seguridad, aceptación final:

Docker + CI/CD automatizado, rollback plan:

ADR, plantillas vivas, manual técnico y usuario:

Métricas Prometheus, alertas Grafana:

Políticas de auditoría y tickets, retención logs:

* ✅ **Requisitos Mínimos**
* 🚀 **Herramientas Avanzadas (Completas)**

**✅ Requisitos Mínimos**

| **Requisito** | **Descripción / Evidencia** | **Estado** | **Acciones / Mejoras Recomendadas** |
| --- | --- | --- | --- |
| **Laravel** | Framework base detectado en composer.json (^12.0) | ✅ Cumple | Ninguna, está correctamente instalado y estructurado |
| **PHP** | Versión requerida ^8.2 (compatible con Laravel 12) | ✅ Cumple | Ninguna |
| **Workbench (BD)** | Migraciones completas, SQL Workbench y ERD documentado | ✅ Cumple | Incluir imagen PNG del ERD final para documentación completa |
| **Seguridad (JWT, Roles, HTTPS)** | JWT (tymon/jwt-auth), RBAC (spatie/permission), HTTPS parcial | ⚠️ Parcial | Implementar TLS completo en Nginx/Proxy y rotar JWT\_SECRET (expuso .env) |
| **Logs (errores y acciones críticas)** | storage/logs/, Monolog, Auditoría (owen-it/auditing) | ✅ Cumple | Centralizar logs a ELK/Loki para búsqueda avanzada |
| **Base de Datos (Migraciones + ERD)** | Migraciones funcionales + documentación ERD | ✅ Cumple | Generar diagrama visual actualizado (php artisan generate:erd) |
| **Git (Ramas + PR básicas)** | Flujo GitFlow y plantillas de PR | ✅ Cumple | Ninguna |
| **Despliegue (DEV/QA/PROD)** | docker-compose.\*.yml y ambientes separados | ⚠️ Parcial | Falta CD automatizado + rollback (blue/green o canary) |
| **Documentación (Swagger/OpenAPI)** | l5-swagger instalado y docs generadas | ✅ Cumple | Mantener docs sincronizadas en pipeline CI |
| **Observabilidad (Logs básicos)** | Prometheus y Grafana configurados | ✅ Cumple | Añadir reglas de alerta + Alertmanager |
| **Gobernanza (PI y protección de datos)** | LICENSE, privacy.md, PRIVACY\_POLICY.md | ✅ Cumple | Añadir política formal de retención de logs |

**🚀 Herramientas del desarrollo del sistema (completas)**

| **Herramienta / Aspecto** | **Descripción / Evidencia** | **Estado** | **Acciones / Mejoras Recomendadas** |
| --- | --- | --- | --- |
| **LDAP + OAuth2 + SSO** | No hay integración LDAP / OAuth2 / SSO | ❌ Falta | Implementar adldap2/adldap2-laravel o laravel/passport y considerar Keycloak o Azure AD |
| **Rate Limiting** | throttle:api básico | ⚠️ Parcial | Migrar a Redis y usar RateLimiter::for() con límites por usuario/IP |
| **Pentesting / Seguridad Avanzada** | Sin scripts o auditorías documentadas | ❌ Falta | Añadir pruebas de pentest, composer audit, y dependabot |
| **Logs estructurados / ELK Stack** | Logs locales y auditorías, sin ELK | ⚠️ Parcial | Integrar ELK o Loki con Monolog ElasticsearchHandler |
| **Auditoría BD / Triggers / Optimización** | owen-it/auditing y triggers en migraciones | ⚠️ Parcial | Implementar Redis para concurrencia alta |
| **Flujo profesional (feature/release/hotfix)** | GitFlow documentado + PR Template | ✅ Cumple | Asegurar code review obligatorio en CI |
| **Pruebas (Carga, Estrés, Aceptación Final)** | No hay scripts (JMeter/k6) | ❌ Falta | Crear pruebas de carga y seguridad en CI/CD |
| **Docker + CI/CD + Rollback** | Docker y CI (build/test) presentes | ⚠️ Parcial | Añadir despliegue automatizado y rollback plan en pipeline |
| **ADR / Documentación Técnica / Manuales** | ADRs, manual admin y usuario | ✅ Cumple | Actualizar manual técnico con últimas versiones |
| **Métricas Prometheus + Alertas Grafana** | Prometheus/Grafana activos, sin alertas | ⚠️ Parcial | Configurar alert.rules.yml y alertmanager.yml |
| **Auditoría / Tickets / Retención Logs** | Políticas básicas de privacidad | ⚠️ Parcial | Definir política formal de retención y crear flujo de tickets internos |

**🔥 Riesgos Críticos (Alta Prioridad)**

| **Riesgo** | **Descripción** | **Acción Inmediata** |
| --- | --- | --- |
| 🔓 **.env expuesto** | Archivos .env incluidos en ZIP/repositorio con credenciales | Eliminar del historial (git filter-repo), rotar todas las claves y usar Secret Manager |
| ⚠️ **Contraseñas en docker-compose** | Variables como MYSQL\_ROOT\_PASSWORD: rootpassword visibles | Sustituir por variables de entorno seguras o docker secrets |
| 🔒 **HTTPS incompleto** | Middleware activa HTTPS, pero Nginx sin bloque SSL | Configurar TLS completo con Let's Encrypt o reverse proxy |
| 🚨 **Sin alertas Prometheus** | Métricas sin reglas de alerta | Crear alert.rules.yml y configurar Alertmanager |
| ⚙️ **CI/CD sin rollback** | Despliegue manual sin rollback plan | Añadir pipeline de deploy + rollback automático en fallos |

**🧭 Roadmap de Implementación (Prioridades)**

| **Plazo** | **Acciones Clave** | **Prioridad** |
| --- | --- | --- |
| ⏰ **48h (Inmediato)** | Eliminar .env del repo, rotar claves, configurar TLS, limpiar contraseñas hardcoded | 🔴 Alta |
| 🗓️ **1–2 semanas** | Implementar Redis (cache/queue), agregar Alertmanager y alert rules, CI/CD con rollback | 🟠 Media-Alta |
| 📅 **1–2 meses** | SSO/LDAP (Keycloak), ELK Stack, pruebas de carga/pentest, centralización de logs | 🟢 Media |