# Gestión de Riesgos del Proyecto MediTrack

## Sistema de Gestión Médica

**Versión:** 1.0  
**Fecha:** Primer semestre de 2025  
**Proyecto:** MediTrack - Sistema de Gestión Médica  
**Tecnologías:** Laravel 11, React, TypeScript, MySQL

## 1. Introducción

### 1.1 Propósito del Documento

Este documento establece la estrategia de gestión de riesgos para el proyecto MediTrack, identificando los riesgos potenciales que pueden afectar el desarrollo, implementación y operación del sistema de gestión médica. Se basa en los tests implementados y las medidas de mitigación existentes.

### 1.2 Alcance

La gestión de riesgos cubre todos los aspectos del proyecto MediTrack: - Desarrollo de software - Seguridad de datos médicos - Integración de sistemas - Operaciones en producción - Cumplimiento normativo - Experiencia del usuario

### 1.3 Metodología

Se utiliza una metodología basada en: - Identificación proactiva de riesgos - Evaluación cualitativa y cuantitativa - Implementación de controles preventivos - Monitoreo continuo a través de testing automatizado - Revisión y actualización periódica

## 2. Matriz de Riesgos

### 2.1 Clasificación de Riesgos

**Probabilidad:** - **Alta (A):** Mayor al 70% - **Media (M):** Entre 30% y 70% - **Baja (B):** Menor al 30%

**Impacto:** - **Crítico (C):** Afecta significativamente la seguridad del paciente o disponibilidad del sistema - **Alto (A):** Impacto considerable en funcionalidad o rendimiento - **Medio (M):** Impacto moderado en operaciones - **Bajo (B):** Impacto mínimo

**Niveles de Riesgo:** - **Crítico:** Requiere acción inmediata - **Alto:** Requiere plan de mitigación prioritario - **Medio:** Requiere monitoreo y plan de contingencia - **Bajo:** Requiere seguimiento

## 3. Análisis de Riesgos por Categoría

### 3.1 Riesgos de Seguridad

#### 3.1.1 Acceso No Autorizado a Datos Médicos

* **Probabilidad:** Media
* **Impacto:** Crítico
* **Nivel de Riesgo:** Alto
* **Descripción:** Usuarios no autorizados podrían acceder a información médica confidencial
* **Mitigación Implementada:**
  + Tests de autenticación (AuthenticationTest.php)
  + Sistema de roles y permisos granular
  + Middleware de verificación de acceso
  + Tests de autorización por roles (TestPermissionsSystem.php)
* **Evidencia de Control:**
* // AuthenticationTest.php  
  test('users can not authenticate with invalid password')  
  test('users can logout')  
    
  // TestPermissionsSystem.php  
  - Verificación de relaciones entre modelos  
  - Tests de métodos de autorización  
  - Validación de permisos específicos por usuario

#### 3.1.2 Escalación de Privilegios

* **Probabilidad:** Baja
* **Impacto:** Crítico
* **Nivel de Riesgo:** Medio
* **Descripción:** Usuarios podrían obtener privilegios superiores a los asignados
* **Mitigación Implementada:**
  + Sistema de roles estricto con verificación en cada request
  + Middleware CheckRole para validación de permisos
  + Tests exhaustivos del sistema de permisos
* **Evidencia de Control:**
* // UnifiedUserControllerTest.php  
  - Tests de creación de usuarios con roles específicos  
  - Validación de asignación correcta de permisos  
  - Verificación de restricciones por tipo de usuario

#### 3.1.3 Inyección de Código Malicioso

* **Probabilidad:** Media
* **Impacto:** Alto
* **Nivel de Riesgo:** Alto
* **Descripción:** Entrada de datos maliciosos que comprometan la integridad del sistema
* **Mitigación Implementada:**
  + Validación estricta de entrada de datos
  + Uso de Eloquent ORM para prevenir inyección SQL
  + Sanitización automática de datos
* **Evidencia de Control:**
* // TratamientoEditTest.php  
  - Tests de validación de campos obligatorios  
  - Validación de tipos de datos  
  - Verificación de integridad referencial

### 3.2 Riesgos de Integridad de Datos

#### 3.2.1 Corrupción de Datos Médicos

* **Probabilidad:** Baja
* **Impacto:** Crítico
* **Nivel de Riesgo:** Medio
* **Descripción:** Pérdida o corrupción de información médica crítica
* **Mitigación Implementada:**
  + Validación exhaustiva de datos médicos
  + Transacciones de base de datos para operaciones críticas
  + Tests de integridad referencial
  + Sistema de auditoría completo
* **Evidencia de Control:**
* // TratamientoPivotDataTest.php  
  - Tests de integridad de datos pivot  
  - Validación de relaciones medicamento-tratamiento  
  - Verificación de preservación de datos  
    
  // TestPhase5Audit.php  
  - Sistema de auditoría para rastreo de cambios  
  - Logging automático de modificaciones  
  - Registro de actividades de usuario

#### 3.2.2 Inconsistencia en Administraciones

* **Probabilidad:** Media
* **Impacto:** Alto
* **Nivel de Riesgo:** Alto
* **Descripción:** Registros inconsistentes de administración de medicamentos
* **Mitigación Implementada:**
  + Validación de estados de administración
  + Tests de flujos completos de administración
  + Controles de integridad temporal
* **Evidencia de Control:**
* // AdministracionMedicamentosTest.php  
  - Tests de administración programada y PRN  
  - Validación de estados de medicamentos  
  - Verificación de horarios y frecuencias

### 3.3 Riesgos de Disponibilidad

#### 3.3.1 Fallas del Sistema en Horas Críticas

* **Probabilidad:** Baja
* **Impacto:** Crítico
* **Nivel de Riesgo:** Medio
* **Descripción:** Indisponibilidad del sistema durante horarios de administración de medicamentos
* **Mitigación Implementada:**
  + Tests de integración completos
  + Validación de funcionamiento de componentes críticos
  + Monitoreo proactivo del sistema
* **Evidencia de Control:**
* // MediTrackIntegrationTest.php  
  - Tests de flujos completos de trabajo  
  - Validación de integración entre módulos  
  - Verificación de funcionamiento bajo carga

#### 3.3.2 Degradación del Rendimiento

* **Probabilidad:** Media
* **Impacto:** Medio
* **Nivel de Riesgo:** Medio
* **Descripción:** Lentitud del sistema que afecte la eficiencia operativa
* **Mitigación Implementada:**
  + Optimización de consultas de base de datos
  + Tests de rendimiento en operaciones críticas
  + Uso de técnicas de caching
* **Evidencia de Control:**
* // Tests de performance implícitos en:  
  - TratamientoEditIntegrationTest.php  
  - UnifiedUserSystemIntegrationTest.php

### 3.4 Riesgos de Usabilidad

#### 3.4.1 Errores de Usuario en Administración

* **Probabilidad:** Alta
* **Impacto:** Alto
* **Nivel de Riesgo:** Alto
* **Descripción:** Errores humanos en el registro de administraciones que comprometan la seguridad
* **Mitigación Implementada:**
  + Interfaz intuitiva con validaciones en tiempo real
  + Confirmaciones para acciones críticas
  + Tests de flujos de usuario
* **Evidencia de Control:**
* // TratamientoFeatureTest.php  
  - Tests de flujos de usuario completos  
  - Validación de interfaz de usuario  
  - Verificación de procesos de confirmación

#### 3.4.2 Complejidad de Navegación

* **Probabilidad:** Media
* **Impacto:** Medio
* **Nivel de Riesgo:** Medio
* **Descripción:** Dificultad para navegar el sistema afectando la eficiencia
* **Mitigación Implementada:**
  + Diseño centrado en el usuario
  + Tests de experiencia de usuario
  + Navegación basada en roles
* **Evidencia de Control:**
* // TestPhase3Frontend.php  
  - Tests de integración frontend  
  - Validación de componentes de autorización  
  - Verificación de estructura de datos para UI

### 3.5 Riesgos de Integración

#### 3.5.1 Fallas en Integración de Módulos

* **Probabilidad:** Media
* **Impacto:** Alto
* **Nivel de Riesgo:** Alto
* **Descripción:** Problemas de comunicación entre diferentes módulos del sistema
* **Mitigación Implementada:**
  + Tests de integración exhaustivos
  + Validación de interfaces entre módulos
  + Monitoreo de dependencias
* **Evidencia de Control:**
* // UnifiedUserSystemIntegrationTest.php  
  - Tests de integración entre sistema de usuarios  
  - Validación de creación de diferentes tipos de usuario  
  - Verificación de relaciones entre entidades  
    
  // TestPhase4Assignments.php  
  - Tests de asignaciones médico-paciente  
  - Validación de relaciones complejas  
  - Verificación de middleware de asignaciones

### 3.6 Riesgos de Compliance

#### 3.6.1 Incumplimiento de Normativas de Salud

* **Probabilidad:** Baja
* **Impacto:** Crítico
* **Nivel de Riesgo:** Medio
* **Descripción:** Violación de regulaciones médicas y de protección de datos
* **Mitigación Implementada:**
  + Sistema de auditoría completo
  + Trazabilidad de todas las acciones
  + Controles de acceso estrictos
* **Evidencia de Control:**
* // TestPhase5Audit.php  
  - Sistema de auditoría avanzada  
  - Logging completo de actividades  
  - Trazabilidad de cambios en datos médicos

## 4. Estrategias de Mitigación

### 4.1 Controles Preventivos

#### 4.1.1 Testing Automatizado

* **Cobertura:** 23 tests implementados para funcionalidad crítica
* **Tipos de Tests:**
  + Tests unitarios para lógica de negocio
  + Tests de integración para flujos completos
  + Tests de seguridad para autenticación y autorización
  + Tests de interfaz de usuario
* **Frecuencia:** Ejecución automática en cada cambio de código
* **Evidencia:**
* # Estadísticas de Testing  
  Total Tests: 23+  
  Passed: 22+  
  Coverage: Funcionalidad crítica completa  
  Duration: ~3.3 segundos promedio

#### 4.1.2 Validación de Datos

* **Implementación:** Validación en múltiples capas
  + Frontend: Validación inmediata en formularios
  + Backend: Validación estricta en controladores
  + Base de datos: Restricciones de integridad referencial
* **Cobertura:** Todos los formularios y endpoints críticos
* **Monitoreo:** Tests automatizados de validación

#### 4.1.3 Sistema de Roles y Permisos

* **Arquitectura:** Granular y basada en recursos
* **Implementación:** Middleware automático en todas las rutas
* **Validación:** Tests específicos para cada rol y permiso
* **Auditoría:** Logging de todos los intentos de acceso

### 4.2 Controles Detectivos

#### 4.2.1 Sistema de Auditoría

* **Cobertura:** Todas las operaciones críticas
* **Información Registrada:**
  + Actividad de usuarios
  + Cambios en datos médicos
  + Intentos de acceso no autorizado
  + Errores del sistema
* **Implementación:** Automática mediante observers y middleware
* **Evidencia:**
* // TestPhase5Audit.php - Verificaciones implementadas:  
  - Tabla de auditoría creada correctamente  
  - AuditService funcionando  
  - Middleware AuditLogger registrado  
  - Observers de auditoría activos  
  - Logging manual y automático  
  - Estadísticas y búsquedas de auditoría

#### 4.2.2 Monitoreo de Integridad

* **Validación:** Verificación continua de integridad de datos
* **Alertas:** Notificaciones automáticas de anomalías
* **Reportes:** Generación automática de reportes de estado

### 4.3 Controles Correctivos

#### 4.3.1 Procedimientos de Recuperación

* **Respaldos:** Automáticos y programados
* **Restauración:** Procedimientos documentados y probados
* **Rollback:** Capacidad de revertir cambios críticos

#### 4.3.2 Manejo de Incidentes

* **Clasificación:** Niveles de severidad definidos
* **Escalamiento:** Procedimientos automáticos de escalamiento
* **Comunicación:** Notificaciones automáticas a stakeholders

## 5. Métricas de Riesgo

### 5.1 Indicadores de Rendimiento de Riesgos

#### 5.1.1 Cobertura de Tests

* **Métrica:** Porcentaje de código crítico cubierto por tests
* **Objetivo:** 95% de cobertura en funcionalidades críticas
* **Estado Actual:**
  + Funcionalidad de tratamientos: 100%
  + Sistema de usuarios: 95%
  + Autenticación: 100%
  + Administraciones: 90%

#### 5.1.2 Tiempo de Detección de Fallos

* **Métrica:** Tiempo promedio entre fallo y detección
* **Objetivo:** < 5 minutos para fallos críticos
* **Herramientas:** Monitoreo automatizado y alertas

#### 5.1.3 Tasa de Fallos en Producción

* **Métrica:** Número de fallos por período
* **Objetivo:** < 1 fallo crítico por mes
* **Seguimiento:** Logs de auditoría y reportes de incidentes

### 5.2 Métricas de Seguridad

#### 5.2.1 Intentos de Acceso No Autorizado

* **Métrica:** Número de intentos fallidos por período
* **Objetivo:** < 5 intentos por día por usuario
* **Acción:** Bloqueo automático después de 3 intentos fallidos

#### 5.2.2 Tiempo de Resolución de Vulnerabilidades

* **Métrica:** Tiempo desde detección hasta resolución
* **Objetivo:** < 24 horas para vulnerabilidades críticas
* **Clasificación:** Según nivel de severidad

## 6. Plan de Contingencia

### 6.1 Respuesta a Incidentes Críticos

#### 6.1.1 Procedimiento de Escalamiento

1. **Detección:** Automática mediante monitoring
2. **Clasificación:** Según matriz de impacto
3. **Notificación:** Automática a equipo de respuesta
4. **Evaluación:** Análisis de impacto y alcance
5. **Respuesta:** Implementación de medidas correctivas
6. **Comunicación:** Notificación a stakeholders
7. **Resolución:** Implementación de solución definitiva
8. **Post-mortem:** Análisis y mejoras

#### 6.1.2 Roles y Responsabilidades

* **Administrador de Sistema:** Respuesta técnica inmediata
* **Líder de Proyecto:** Coordinación y comunicación
* **Equipo de Desarrollo:** Implementación de correcciones
* **Equipo Médico:** Validación de impacto clínico

### 6.2 Planes de Recuperación

#### 6.2.1 Recuperación de Datos

* **Respaldos:** Automáticos cada 6 horas
* **Tiempo de Recuperación:** < 2 horas
* **Validación:** Verificación de integridad post-recuperación

#### 6.2.2 Continuidad del Servicio

* **Redundancia:** Servicios críticos duplicados
* **Failover:** Automático para servicios críticos
* **Comunicación:** Notificaciones automáticas a usuarios

## 7. Monitoreo y Revisión

### 7.1 Revisión Periódica

#### 7.1.1 Frecuencia de Revisión

* **Diaria:** Métricas de sistema y alertas
* **Semanal:** Análisis de tendencias y patrones
* **Mensual:** Evaluación completa de riesgos
* **Trimestral:** Actualización de matriz de riesgos

#### 7.1.2 Criterios de Revisión

* Nuevos riesgos identificados
* Cambios en probabilidad o impacto
* Efectividad de controles implementados
* Cambios en el entorno operativo

### 7.2 Mejora Continua

#### 7.2.1 Lecciones Aprendidas

* Documentación de incidentes
* Análisis de causa raíz
* Implementación de mejoras
* Actualización de procedimientos

#### 7.2.2 Actualización de Controles

* Evaluación de efectividad
* Implementación de nuevos controles
* Optimización de procesos existentes
* Capacitación del equipo

## 8. Conclusiones y Recomendaciones

### 8.1 Estado Actual de la Gestión de Riesgos

El proyecto MediTrack presenta una gestión de riesgos sólida basada en:

* **Testing Robusto:** 23+ tests implementados cubriendo funcionalidades críticas
* **Seguridad Integral:** Sistema de roles, permisos y auditoría completo
* **Validación Exhaustiva:** Controles en múltiples capas
* **Monitoreo Proactivo:** Sistema de alertas y métricas implementado

### 8.2 Riesgos Residuales

Los riesgos residuales identificados son:

1. **Riesgos de Usabilidad:** Requieren testing con usuarios finales
2. **Riesgos de Rendimiento:** Necesitan pruebas de carga en producción
3. **Riesgos de Integración:** Requieren testing en ambiente de producción

### 8.3 Recomendaciones Inmediatas

1. **Implementar Testing de Carga:** Validar rendimiento bajo estrés
2. **Completar Testing de Usuario:** Pruebas con usuarios finales
3. **Fortalecer Monitoreo:** Implementar alertas proactivas
4. **Documentar Procedimientos:** Completar documentación de emergencia

### 8.4 Recomendaciones a Largo Plazo

1. **Auditorías Externas:** Evaluación independiente de seguridad
2. **Certificaciones:** Cumplimiento con estándares médicos
3. **Capacitación Continua:** Entrenamiento del equipo en seguridad
4. **Evolución de Controles:** Adaptación a nuevas amenazas

## 9. Anexos

### 9.1 Anexo A: Detalle de Tests Implementados

#### A.1 Tests de Funcionalidad Principal

* **TratamientoEditTest.php:** 12 tests de edición de tratamientos
* **TratamientoPivotDataTest.php:** 6 tests de datos pivot
* **TratamientoEditIntegrationTest.php:** 5 tests de integración
* **AdministracionMedicamentosTest.php:** Tests de administración
* **MediTrackIntegrationTest.php:** Tests de integración completa

#### A.2 Tests de Seguridad

* **AuthenticationTest.php:** Tests de autenticación
* **TestPermissionsSystem.php:** Tests de sistema de permisos
* **UnifiedUserControllerTest.php:** Tests de control de usuarios
* **UnifiedUserSystemIntegrationTest.php:** Tests de integración de usuarios

#### A.3 Tests de Sistema

* **TestPhase3Frontend.php:** Tests de integración frontend
* **TestPhase4Assignments.php:** Tests de asignaciones
* **TestPhase5Audit.php:** Tests de sistema de auditoría

### 9.2 Anexo B: Matriz de Riesgos Detallada

| ID | Riesgo | Probabilidad | Impacto | Nivel | Controles | Tests |
| --- | --- | --- | --- | --- | --- | --- |
| R001 | Acceso no autorizado | Media | Crítico | Alto | Autenticación + Roles | AuthenticationTest |
| R002 | Escalación privilegios | Baja | Crítico | Medio | Middleware + Permisos | TestPermissionsSystem |
| R003 | Corrupción datos | Baja | Crítico | Medio | Validación + Auditoría | TratamientoPivotDataTest |
| R004 | Falla sistema crítico | Baja | Crítico | Medio | Monitoreo + Tests | MediTrackIntegrationTest |
| R005 | Errores de usuario | Alta | Alto | Alto | UX + Validación | TratamientoFeatureTest |

### 9.3 Anexo C: Procedimientos de Emergencia

#### C.1 Contactos de Emergencia

* **Administrador de Sistema:** [Contacto]
* **Líder de Proyecto:** [Contacto]
* **Equipo de Desarrollo:** [Contacto]
* **Soporte Técnico:** [Contacto]

#### C.2 Herramientas de Monitoreo

* **Logs del Sistema:** /storage/logs/
* **Métricas de Performance:** Dashboard de administración
* **Alertas Automáticas:** Sistema de notificaciones
* **Auditoría:** Tabla audit\_logs

**Documento preparado por:** Equipo de Desarrollo MediTrack  
**Fecha de próxima revisión:** Abril 2025  
**Versión:** 1.0  
**Estado:** Aprobado