

# Gestión de Riesgos - Sistema de Gestión de Inventarios

## Información del Proyecto

- **Proyecto:** Sistema de Gestión de Inventarios con QAS
- **Fecha:** Agosto 2025
- **Responsable:** Randae Garcia
- **Revisión:** v1.0

## 1. Metodología de Gestión de Riesgos

### 1.1 Proceso de Identificación

- Análisis de proyectos similares y lecciones aprendidas
- Revisión de documentación técnica y requisitos
- Evaluación de dependencias tecnológicas



### 1.2 Matriz de Evaluación




Probabilidad	Descripción	Valor
Muy Alta	> 80% de ocurrencia	5
Alta	60-80% de ocurrencia	4
Media	40-60% de ocurrencia	3
Baja	20-40% de ocurrencia	2

Muy Baja	< 20% de ocurrencia	1
----------	---------------------	---

Impacto	Descripción	Valor
Crítico	Paraliza el proyecto completamente	5
Alto	Retraso significativo (>2 semanas)	4
Medio	Retraso moderado (1-2 semanas)	3
Bajo	Retraso menor (<1 semana)	2
Mínimo	Sin impacto en cronograma	1

**Nivel de Riesgo = Probabilidad × Impacto**


Puntuación	Nivel	Color	Acción
20-25	Crítico		Acción inmediata
15-19	Alto		Plan de mitigación urgente

10-14	Medio		Monitoreo activo
5-9	Bajo		Monitoreo periódico
1-4	Mínimo		Documentar únicamente

## 2. Registro de Riesgos Identificados

### 2.1 Riesgos Técnicos

#### RT-001: Incompatibilidad entre Versiones de Tecnologías

- **Descripción:** Conflictos entre versiones de Quarkus, Vue.js, Keycloak y dependencias
- **Probabilidad:** 3 (Media)
- **Impacto:** 4 (Alto)
- **Nivel de Riesgo:** 12 
- **Categoría:** Técnico
- **Disparador:** Actualizaciones de dependencias, cambios de versión


#### Plan de Mitigación:

- Usar versiones LTS y estables documentadas
- Implementar tests de integración automatizados
- Documentar versiones exactas en Docker

#### Plan de Contingencia:

- Rollback a versiones anteriores conocidas
- Usar contenedores Docker con versiones específicas
- Búsqueda de alternativas tecnológicas

## RT-002: Problemas de Conectividad con Keycloak

- **Descripción:** Fallos en la autenticación/autorización por problemas de red o configuración
- **Probabilidad:** 3 (Media)
- **Impacto:** 5 (Crítico)
- **Nivel de Riesgo:** 15 
- **Categoría:** Técnico/Seguridad
- **Disparador:** Cambios de red, configuración incorrecta, caída de servicios

### Plan de Mitigación:


- Implementar health checks para Keycloak
- Configurar timeouts y reintentos automáticos
- Documentar configuración de red requerida

### Plan de Contingencia:

- Activar modo offline temporal
- Usar autenticación básica de emergencia
- Levantar instancia de Keycloak alternativa

---

## RT-003: Pérdida de Datos por Fallos en Migración

- **Descripción:** Corrupción o pérdida de datos durante migraciones de Flyway
- **Probabilidad:** 2 (Baja)
- **Impacto:** 5 (Crítico)
- **Nivel de Riesgo:** 10 
- **Categoría:** Datos
- **Disparador:** Errores en scripts SQL, interrupciones durante migración

### Plan de Mitigación:


- Backup automático antes de cada migración
- Validar scripts en ambiente de testing
- Implementar rollback automático en caso de error

- Versionado de esquemas de base de datos

#### **Plan de Contingencia:**

- Restaurar desde backup más reciente
- Ejecutar scripts de reparación manual
- Reconstruir base de datos desde cero si es necesario

#### **RT-004: Problemas de Performance con Grandes Volúmenes**

- **Descripción:** Degradación del rendimiento con alta cantidad de productos y movimientos
- **Probabilidad:** 4 (Alta)
- **Impacto:** 3 (Medio)
- **Nivel de Riesgo:** 12 
- **Categoría:** Performance
- **Disparador:** Crecimiento de datos, consultas no optimizadas

#### **Plan de Mitigación:**

- Implementar paginación en todos los listados
- Crear índices en campos frecuentemente consultados
- Usar caché para consultas repetitivas
- Monitorear métricas de performance

#### **Plan de Contingencia:**

- Optimizar consultas específicas problemáticas
- Implementar caché adicional
- Escalar horizontalmente la base de datos

## **2.2 Riesgos de Proyecto**

#### **RP-001: Retrasos por Complejidad de Configuración**

- **Descripción:** Tiempo excesivo configurando Docker, Keycloak y integraciones
- **Probabilidad:** 4 (Alta)
- **Impacto:** 3 (Medio)

- **Nivel de Riesgo:** 12 ●
- **Categoría:** Cronograma
- **Disparador:** Falta de experiencia con tecnologías, documentación insuficiente

**Plan de Mitigación:**

- Crear guías paso a paso detalladas
- Automatizar configuración con scripts
- Asignar tiempo extra para configuración inicial
- Tener ambiente de referencia funcionando

**Plan de Contingencia:**

- Priorizar funcionalidades core sobre configuraciones avanzadas
- Simplificar arquitectura si es necesario
- Buscar ayuda externa o mentoría

**RP-002: Falta de Conocimiento en Tecnologías Específicas**

- **Descripción:** Curva de aprendizaje alta en Quarkus, Vue.js 3, o Keycloak
- **Probabilidad:** 3 (Media)
- **Impacto:** 4 (Alto)
- **Nivel de Riesgo:** 12 ●
- **Categoría:** Recursos Humanos
- **Disparador:** Implementación de funcionalidades complejas

**Plan de Mitigación:**


- Dedicar tiempo inicial a documentación
- Crear prototipos simples para practicar
- Consultar documentación oficial y comunidades
- Implementar incrementalmente

**Plan de Contingencia:**

- Cambiar a tecnologías más familiares si es crítico
- Reducir scope de funcionalidades avanzadas
- Buscar ejemplos y tutoriales adicionales

## 2.3 Riesgos de Calidad

### RC-001: Cobertura Insuficiente de Pruebas

- **Descripción:** No alcanzar los estándares de testing requeridos para QAS
- **Probabilidad:** 3 (Media)
- **Impacto:** 4 (Alto)
- **Nivel de Riesgo:** 12 
- **Categoría:** Calidad
- **Disparador:** Presión de tiempo, complejidad de setup de testing

#### Plan de Mitigación:


- Definir métricas mínimas de cobertura (80%)
- Implementar testing desde el inicio del desarrollo
- Usar herramientas automatizadas (JaCoCo, Playwright)
- Priorizar testing de funcionalidades críticas

#### Plan de Contingencia:

- Enfocarse en testing manual exhaustivo
- Implementar pruebas de aceptación claras
- Documentar casos de prueba no automatizados

## 2.4 Riesgos Externos

### RE-001: Cambios en Requisitos del Proyecto

- **Descripción:** Modificaciones significativas en scope o funcionalidades requeridas
- **Probabilidad:** 2 (Baja)
- **Impacto:** 4 (Alto)
- **Nivel de Riesgo:** 8 
- **Categoría:** Scope
- **Disparador:** Feedback de stakeholders, cambios académicos

#### Plan de Mitigación:

- Documentar requisitos claramente desde el inicio

- Implementar funcionalidades en iteraciones
- Mantener comunicación constante con stakeholders
- Crear arquitectura flexible

**Plan de Contingencia:**

- Implementar cambios como mejoras futuras
- Re-priorizar funcionalidades existentes