

# **Sistema de gestión de inventario - Pool Center**

## **Informe de Revisión técnica formal (RTF)**

**Versión [1.0]**

### **Historia de revisiones**

Fecha	Versión	Descripción	Autor
15/08/2025	1.0	Primera versión del informe de revisión técnica formal	Grupo 4

## Contenido

<b>SISTEMA DE GESTIÓN DE INVENTARIO - POOL CENTER</b>	<b>1</b>
<b>INFORME DE REVISIÓN TÉCNICA FORMAL (RTF)</b>	<b>1</b>
<b>VERSIÓN [1.0]</b>	<b>1</b>
<b>HISTORIA DE REVISIONES</b>	<b>1</b>
<b>1. PRODUCTO REVISADO</b>	<b>3</b>
1.1. NOMBRE Y VERSIÓN DEL PRODUCTO REVISADO	3
1.2. PARTICIPANTES DE LA REVISIÓN	3
1.3. TÉCNICA UTILIZADA	3
<b>2. OBJETIVOS DE LA RTF</b>	<b>3</b>
<b>3. PROBLEMAS DETECTADOS</b>	<b>3</b>
3.1. PROBLEMA DETECTADO 1	3
3.1.1. <i>Sugerencia de corrección</i>	3
3.2. PROBLEMA DETECTADO 2No se IMPLEMENTAN BACKUPS AUTOMÁTICOS EN LA BASE DE DATOS, SOLO SE CUENTA CON ROLLBACKS.SEVERIDAD: MODERADO.UBICACIÓN: BASE DE DATOS (POSTGRESQL).	3
3.2.1. <i>Sugerencia de corrección</i>	4
3.3. PROBLEMA DETECTADO 3No existe un ARCHIVO. ENV.EXAMPLE PARA ORIENTAR LA CONFIGURACIÓN EN NUEVOS ENTORNOS.SEVERIDAD: MENOR.UBICACIÓN: SEGURIDAD/CONFIGURACIÓN.	4
3.3.1. <i>Sugerencia de corrección</i> Crear un archivo. env.example con las variables mínimas requeridas y descripciones; automatizar la validación de variables faltantes en arranque.	4
3.4. PROBLEMA DETECTADO 4En ciertas resoluciones pequeñas, algunos elementos no se adaptan correctamente y llegan a traslaparse.SEVERIDAD: MODERADO.UBICACIÓN: FRONTEND / RESPONSIVE	4
3.4.1. <i>Sugerencia de corrección</i>	4
3.5. PROBLEMA DETECTADO 5	4
3.5.1. <i>Sugerencia de corrección</i> Optimizar las imágenes o utilizar una API; implementar lazy loading para mejorar tiempos de carga y asegurar rutas correctas en el renderizado	4
<b>4. EVALUACIÓN</b>	<b>4</b>
4.1. ESTADO ACTUAL DEL PRODUCTO	4
4.2. ACCIONES A TOMAR	5
4.3. PRÓXIMA REVISIÓN DEL PRODUCTO	5

## **1. Producto revisado**

### **1.1. Nombre y Versión del Producto revisado**

Nombre: PoolCenter - Sistema de gestión de Inventario

Versión: 1.0

Área: Desarrollo Web (Frontend, Backend, Base de Datos)

### **1.2. Participantes de la revisión**

Participantes:

- Iris Ayala (Ingeniera)
- David Domínguez (Ingeniero)
- Anggie Quezada (Ingeniera)
- Gabriel Bran (Inspector)
- Luis Padilla (Inspector)
- Carlos Alburez (Revisor)
- Kevin Villagrán (Revisor)
- Erick Guerra (Revisor)

### **1.3. Técnica utilizada**

Se utilizó una lista de cotejo definida en el Plan de SQA, donde se revisaron módulos de frontend, backend, base de datos y seguridad. Se validaron aspectos de arquitectura, usabilidad, pruebas, documentación y medidas de seguridad. La revisión incluyó sesiones de inspección en equipo y uso de checklists compartidos.

## **2. Objetivos de la RTF**

Los objetivos de la RTF fueron:

- Verificar que los módulos de marketing e inventario cumplan principios de arquitectura limpia.
- Contar con documentación técnica y funcional.
- Validar usabilidad, diseño y rendimiento en marketing.
- Comprobar exactitud y seguridad en inventario.
- Reducir defectos críticos a menos de 5.
- Asegurar la preparación de los revisores.

## **3. Problemas detectados**

Durante la inspección se detectaron los siguientes problemas:

### **3.1. Problema detectado 1**

Falta de documentación de endpoints en plataforma estándar (Swagger/Postman).

Severidad: Moderado.

Ubicación: Backend.

#### **3.1.1. Sugerencia de corrección**

Sugerencia: Documentar todos los endpoints utilizando Swagger o Postman de manera clara y accesible.

### **3.2. Problema detectado 2**

No se implementan backups automáticos en la base de datos, solo se cuenta con rollbacks.

Severidad: Moderado.  
Ubicación: Base de Datos (PostgreSQL).

#### **3.2.1. Sugerencia de corrección**

Configurar backups automáticos (pg\_dump programado y verificado), retención por políticas y pruebas periódicas de restauración; considerar snapshots o servicios administrados.

### **3.3. Problema detectado 3**

No existe un archivo. env.example para orientar la configuración en nuevos entornos.  
Severidad: Menor.  
Ubicación: Seguridad/Configuración.

#### **3.3.1. Sugerencia de corrección**

Crear un archivo. env.example con las variables mínimas requeridas y descripciones; automatizar la validación de variables faltantes en arranque.

### **3.4. Problema detectado 4**

En ciertas resoluciones pequeñas, algunos elementos no se adaptan correctamente y llegan a traslaparse.  
Severidad: Moderado.  
Ubicación: Frontend / Responsive.

#### **3.4.1. Sugerencia de corrección**

Ajustar el diseño; probar en múltiples tamaños de pantalla y dispositivos; aplicar un sistema de rejilla flexible para evitar traslapes.

### **3.5. Problema detectado 5**

Algunas imágenes no se renderizan correctamente en el frontend, afectando la experiencia del usuario.

Severidad: Severa.  
Ubicación: Frontend / Renderizado de imágenes.

#### **3.5.1. Sugerencia de corrección**

Optimizar las imágenes o utilizar una API; implementar lazy loading para mejorar tiempos de carga y asegurar rutas correctas en el renderizado

## **4. Evaluación**

### **4.1. Estado actual del Producto**

Estado actual del Producto: El producto es funcional y cumple en gran medida con los criterios de calidad. No presenta defectos críticos ni severos, pero requiere correcciones moderadas (documentación API, backups y optimización de recursos).

#### **4.2. Acciones a tomar**

- Implementar documentación de API con Swagger/Postman
- Configurar backups automáticos en la base de datos.
- Incluir un archivo .env.example para nuevas configuraciones.
- Mejorar aspectos en el frontend y API para imágenes.
- Revisar nuevamente el producto tras estas mejoras.

#### **4.3. Próxima Revisión del Producto**

Próxima Revisión: Programada para el 20/08/2025, posterior a la corrección de los problemas moderados y menores detectados.