

Informe Técnico: Fallos Detectados en la Versión 1 del Esquema de Base de Datos

Proyecto: Sistema de Gestión de Biblioteca Fecha de Análisis: 30/1/2025

1. Introducción:

La versión 1 del esquema de base de datos (Script 2) presenta inconsistencias y deficiencias críticas que incumplen requisitos funcionales, técnicos y de normalización. Este informe detalla los fallos identificados, su impacto en el sistema y propuestas de mejora basadas en la versión 2 (Script 1), que corrige estos problemas.

2. Metodología

Se analizó el esquema de la versión 1 contra los requisitos del enunciado, identificando:

- Incumplimiento de formas normales.
- Falta de validaciones y triggers críticos.
- Deficiencias en la gestión de datos y reportes.

3. Fallos Identificados

3.1. Normalización Insuficiente

Fallo	Impacto	Solución Propuesta
Denormalización en la tabla libros: Columnas autor, genero, categoria almacenan valores repetidos (violación de 1FN). Ejemplo: Un libro con múltiples autores no puede registrarse sin duplicar filas.	Redundancia de datos, inconsistencia en búsquedas y dificultad para actualizar registros.	Implementar tablas relacionales (autores, generos, categorias) y tablas de unión (libros_autores, etc.).
Falta de separación de roles y permisos: El campo rol en usuarios usa ENUM en lugar de una tabla dedicada.	Limitación para escalar roles o asignar permisos granularmente.	Crear tablas roles, permisos y roles_permisos para gestión flexible (5FN).

3.2. Gestión de Préstamos y Reservas

Fallo	Impacto	Solución Propuesta
Triggers incompletos: El trigger actualizar_estado_libro no maneja reservas pendientes al devolver un libro. No hay validación de disponibilidad antes de préstamos.	Libros devueltos no se asignan a reservas pendientes, generando conflictos.	Implementar triggers que actualicen el estado del libro a "reservado" si hay reservas activas.
Falta de estados de reserva: La tabla reservas usa ENUM para estados sin una tabla dedicada.	Dificultad para agregar nuevos estados o realizar seguimiento.	Crear tabla estados_reserva y normalizar la relación (3FN).

### 3.3. Validaciones y Seguridad

Fallo	Impacto	Solución Propuesta
Validación de contactos débil: Email y teléfono se almacenan directamente en usuarios sin normalización.	Posibilidad de duplicar contactos (ej: mismo email en múltiples usuarios).	Separar en tipos_contacto y contactos_usuario con restricciones UNIQUE (4FN).
Contraseñas sin estándares robustos: El campo password no especifica si usa salting o algoritmos modernos.	Vulnerabilidad a ataques de fuerza bruta o filtración de datos.	Usar SHA-256 con salting y almacenar en CHAR(64).

### 3.4. Reportes y Consultas

Fallo	Impacto	Solución Propuesta
Falta de vistas estructuradas: No hay vistas predefinidas para reportes como "libros más prestados" o "porcentaje por género".	Consultas manuales complejas y poco eficientes para generar reportes.	Crear vistas como v_libros_mas_prestados y v_porcentaje_libros_genero.
Datos históricos no normalizados: La tabla historial_prestamos duplica información de prestamos.	Redundancia y riesgo de inconsistencia en datos históricos.	Eliminar redundancias usando triggers y normalización.

### 4. Impacto en Requisitos Funcionales:

La versión 1 incumple los siguientes requisitos del enunciado:

- Gestión de roles granular: No permite asignar permisos específicos a roles.
- Límite de préstamos para usuarios: El trigger limitar\_prestamos\_usuario no valida correctamente el rol del usuario.
- Préstamos express automáticos: La tabla prestamos\_express no se actualiza eficientemente.
- Reservas con seguimiento: No hay transiciones de estado claras (pendiente → completada → vencida).

### 5. Recomendaciones:

Para corregir los fallos, se propone adoptar la versión 2 del esquema (Script 1), que:

- Normaliza datos hasta 5FN con tablas relacionales y de unión.
- Implementa triggers robustos para validar disponibilidad, actualizar estados y gestionar reservas.
- Mejora la seguridad con validaciones de contactos únicos y contraseñas cifradas.
- Facilita reportes mediante vistas predefinidas y consultas optimizadas.

## 6. Conclusión:

La versión 1 presenta deficiencias críticas en normalización, validaciones y gestión de datos, lo que limita la escalabilidad y seguridad del sistema. La versión 2 soluciona estos problemas mediante un diseño relacional robusto y cumplimiento estricto de los requisitos del proyecto.