

Trabajo Final Integrador
Tecnicatura Universitaria en Programación
Universidad Tecnológica Nacional



Materia: Programación II

Profesor: Cinthia Rigoni

Tutor: Jerónimo Felipe Cortéz

Alumnos: Herrera Florencia, Palma Uriel, Pattini Lorenzo,

Quinteros Federico (Grupo 82).

Fecha de entrega: 17/11/2025

Elección del Dominio y Justificación

Se desarrolló un Sistema de Gestión Bibliográfica para administrar Libro (Clase A) y sus FichaBibliografica (Clase B) en una biblioteca.

Es un dominio natural y claro: cada libro tiene una y solo una ficha bibliográfica (ISBN, clasificación Dewey, estantería, etc.), y esa ficha pertenece exclusivamente a ese libro.

Permite modelar una relación 1→1 con restricciones de unicidad claras.

Evita ambigüedades: no tiene sentido que dos libros comparten la misma ficha, ni que una ficha exista sin libro.

Diseño del Sistema

- La relación 1:1 entre Libro y FichaBibliografica cumple con el requisito de asociación uno a uno, permite la nulabilidad opcional para que existan libros sin ficha, mantiene la normalización de la base de datos y facilita la realización de consultas independientes entre ambas entidades.
- Respecto a la eliminación lógica vs física, la eliminación lógica preserva la integridad referencial histórica, permite recuperar datos que fueron eliminados accidentalmente y cumple con los estándares utilizados en sistemas empresariales modernos.

Diagrama UML

Ver UML: [UML](#)

Arquitectura por Capas - Responsabilidades

- **Capa de Entidades (entities)**

La capa de entidades representa el modelo de dominio, contiene las validaciones básicas de datos, implementa la lógica de negocio elemental y está compuesta por POJOs (Plain Old Java Objects).

- **Capa de Acceso a Datos (dao)**

La capa DAO abstrae las operaciones CRUD, implementa la lógica específica de SQL necesaria para la persistencia, realiza el mapeo entre ResultSet y entidades, y participa directamente en las transacciones.

- **Capa de Servicio (service)**

La capa de servicio coordina las operaciones que involucran múltiples DAOs, implementa reglas de negocio más complejas, gestiona las transacciones a nivel de servicio y provee una API coherente para la capa de presentación.

- **Capa de Presentación (main)**

La capa de presentación corresponde a la interfaz de usuario por consola. Se encarga de la captura y validación de datos ingresados por el usuario, maneja la navegación del menú y presenta los resultados de las operaciones realizadas.

Persistencia de Datos

Estructura de la Base de Datos

Índices y Restricciones

- Índices: título, autor, ISBN
- UNIQUE CONSTRAINT: idLIBRO (garantiza 1:1)
- UNIQUE CONSTRAINT: isbn (ISBN único en el sistema)
- TRIGGER: Validación adicional para relación 1:1

Gestión de Transacciones

Estrategia de Transacciones

- Nivel: Transacciones a nivel de servicio
- Alcance: Operaciones que involucran múltiples DAOs
- Control: Commit/Rollback explícito

Orden de Operaciones en Creación

1. Iniciar transacción
2. Insertar Libro (obtener ID generado)
3. Insertar FichaBibliografica (usar ID de Libro)
4. Commit de transacción
5. En caso de error: rollback completo

Validaciones y Reglas de Negocio-

Validaciones de Entidad

Libro

```
public void setTitulo(String titulo) {  
    if (titulo == null || titulo.trim().isEmpty()) {  
        throw new IllegalArgumentException("El titulo no puede ser nulo o vacio");  
    }  
    if (titulo.length() > 150) {  
        throw new IllegalArgumentException("El titulo excede los 150 caracteres");  
    }  
    this.titulo = titulo.trim();  
}
```

```
public void setAnioEdicion(Integer anioEdicion) {  
    if (anioEdicion != null && (anioEdicion < 1450 || anioEdicion > 2025)) {  
        throw new IllegalArgumentException("Anio de edicion fuera de rango valido (1450-2025)");  
    }  
    this.anioEdicion = anioEdicion;  
}
```

FichaBibliografica

```
public void setIsbn(String isbn) {  
    if (isbn != null) {  
        String clean = isbn.replaceAll("[\\s\\-]", ""); // Normalizacion, se quitan espacios y guiones  
        if (clean.length() != 10 && clean.length() != 13) {  
            throw new IllegalArgumentException("ISBN debe tener 10 o 13 dígitos (sin guiones)");  
        }  
        if (!clean.matches("\\d{9}[\\dXx]|\\d{12}[\\dXx]")) {  
            throw new IllegalArgumentException("ISBN contiene caracteres invalidos");  
        }  
        if (isbn.length() > 17) {  
            throw new IllegalArgumentException("ISBN excede los 17 caracteres (con guiones)");  
        }  
    }  
    this.isbn = isbn;  
}
```

Reglas de Negocio

Relación 1:1 Estricta

- Un libro puede tener como máximo una ficha bibliográfica activa
- Una ficha bibliográfica pertenece a exactamente un libro
- Validación a nivel de base de datos mediante CONSTRAINT UNIQUE

Unicidad de ISBN

- El ISBN debe ser único en el sistema
- Validación a nivel de base de datos y aplicación
- Búsqueda case-insensitive

Integridad Referencial

- No se puede crear ficha para libro inexistente
- Eliminación en cascada configurada en la base de datos
- Validación de existencia previa a operaciones

Pruebas Realizadas

Flujo Principal - Creación de Libro con Ficha

```
Bienvenido al Sistema de Gestión Bibliográfica

==== MENU PRINCIPAL ====
1. Crear Libro
2. Buscar Libro por ID
3. Listar todos los Libros activos
4. Buscar Libro por título
5. Actualizar Libro
6. Eliminar Libro
7. Buscar Ficha por ISBN
8. Ver Ficha asociada a un Libro
9. Salir
Elija una opción: 1

--- Crear Libro ---
Título (máx. 150): New World Aeternum
Autor (máx. 120): Amazon Games
Editorial, opcional (máx. 100): AGS
Año de edición, opcional: 2021
Datos de la Ficha Bibliográfica:
ISBN, opcional (10 o 13 dígitos, máx. 17 caracteres): 978-84-376-1111-7
Clasificación Dewey, opcional (máx. 20): 800.54
Estantería, opcional (máx. 20): A3-B12
Idioma, opcional (máx. 30): Inglés
    Libro y Ficha creados con éxito.
ID Libro: 8
ID Ficha: 8
ISBN: 978-84-376-1111-7
Presione ENTER para continuar...

==== MENU PRINCIPAL ====
1. Crear Libro
2. Buscar Libro por ID
3. Listar todos los Libros activos
4. Buscar Libro por título
5. Actualizar Libro
6. Eliminar Libro
7. Buscar Ficha por ISBN
8. Ver Ficha asociada a un Libro
9. Salir
Elija una opción: 2
ID del Libro: 8
Libro encontrado:
Libro{id=8, título='New World Aeternum', autor='Amazon Games', editorial='AGS', añoEdición=2021, eliminado=false}
Ficha: FichaBibliográfica{id=8, isbn='978-84-376-1111-7', clasificaciónDewey='800.54', estantería='A3-B12', idioma='Inglés', eliminado=false}
Presione ENTER para continuar...
```

Pruebas de Validación

Validación de ISBN Inválido

```
==== MENU PRINCIPAL ====
1. Crear Libro
2. Buscar Libro por ID
3. Listar todos los Libros activos
4. Buscar Libro por titulo
5. Actualizar Libro
6. Eliminar Libro
7. Buscar Ficha por ISBN
8. Ver Ficha asociada a un Libro
9. Salir
Elija una opcion: 7

--- Buscar Ficha por ISBN ---
ISBN: 978-84-376-1111-7
Ficha encontrada:
FichaBibliografica(id=8, isbn='978-84-376-1111-7', clasificacionDewey='800.54', estanteria='A3-B12', idioma='Ingl s', eliminado=false)
Presione ENTER para continuar...
```

Validación de Campos Obligatorios

```
run:
Bienvenido al Sistema de Gestión Bibliográfica

==== MENU PRINCIPAL ====
1. Crear Libro
2. Buscar Libro por ID
3. Listar todos los Libros activos
4. Buscar Libro por título
5. Actualizar Libro
6. Eliminar Libro
7. Buscar Ficha por ISBN
8. Ver Ficha asociada a un Libro
9. Salir
Elija una opción: 1

--- Crear Libro ---
Título (max. 150):
Título (max. 150):
Título (max. 150): |
```

Pruebas de Transacciones

Rollback por Violación de UNIQUE

- Intento de crear segunda ficha para mismo libro
- Sistema detecta violación de constraint única
- Rollback automático de toda la transacción

- Mensaje: "El Libro ya tiene una FichaBibliografica asociada"

Conclusiones y Mejoras Futuras

Logros Alcanzados

- Implementación exitosa de arquitectura por capas
- Gestión robusta de transacciones de base de datos
- Relación 1:1 correctamente implementada y validada
- Interfaz de usuario intuitiva y funcional
- Validaciones completas de negocio y datos
- Manejo apropiado de errores y excepciones

Aprendizajes Clave

- La separación en capas facilita el mantenimiento y testing
- Las transacciones son esenciales para operaciones atómicas
- Las validaciones deben implementarse en múltiples niveles
- El diseño de base de datos impacta directamente en la aplicación

Mejoras Futuras

Mejoras Técnicas

1. Implementar pooling de conexiones para mejor rendimiento
2. Agregar logging estructurado con diferentes niveles
3. Implementar paginación para grandes volúmenes de datos
4. Agregar cache para consultas frecuentes

Funcionalidades Adicionales

1. Búsqueda avanzada por múltiples criterios
2. Exportación de datos a formatos estándar (PDF, Excel)
3. Backup y restore de la base de datos
4. Interfaz web además de la consola

Seguridad y Robustez

1. Validación de entrada más exhaustiva
2. Manejo de concurrencia con optimistic locking
3. Auditoría de cambios en registros
4. Configuración externalizada completa

Fuentes y Herramientas Utilizadas

Tecnologías y Herramientas

- Lenguaje: Java 17+
- Base de Datos: MySQL 8.0+
- Driver JDBC: mysql-connector-java 8.0+
- IDE: Visual Studio Code / IntelliJ IDEA
- Control de Versiones: Git
- Diagramas: PlantUML para diagramas UML

Referencias Bibliográficas

1. Oracle Java Documentation
2. MySQL 8.0 Reference Manual

Uso de IA

ChatGPT/DeepSeek fue utilizado como asistente para:

- Revisión de mejores prácticas en arquitectura por capas
- Sugerencias sobre implementación de transacciones
- Ayuda en la redacción de documentación técnica
- Resolución de dudas específicas sobre JDBC y MySQL

Anexo

Repositorio GitHub: <https://github.com/PalmaUriel/TrabajointegradorGrupo82>