LIBRO→FICHABIBLIOGRAFICA

Trabajo Integrador Bases de Datos

Grupo N° 92 – Comisión N°6

Alumnos:
BRIÑÓCCOLI, Adrian Nahuel – email: adrianbrinoccoli@gmail.com
BUJALDON DUARTE, Octavio Carlos – email: octa.buja@gmail.com

Materia:
Programación II

Coordinador:
Carlos Martínez

Profesor:
Ariel Enferrel

Profesor Tutor:
Federico Frankerberger

17 de Noviembre de 2025

Introducción:

1.1: Objetivo Principal:

El presente Trabajo Final Integrador (TFI) tuvo como objetivo principal desarrollar una aplicación de consola en Java capaz de gestionar dos entidades de dominio relacionadas mediantes una asociación 1 a 1 unidireccional, utilizando JDBC (Java Database Connectivity) para la persistencia en una base de datos MySQL.
El proyecto está diseñado bajo una arquitectura de capas rígidas, aplicando el patrón DAO (Data Access Object) y la capa Service, con un énfasis fundamental en la correcta implementación de Transacciones Explicitas (commit/rollback) para garantizar la integridad de los datos en operaciones compuesta.

1.2: Tecnologías y Alcances:

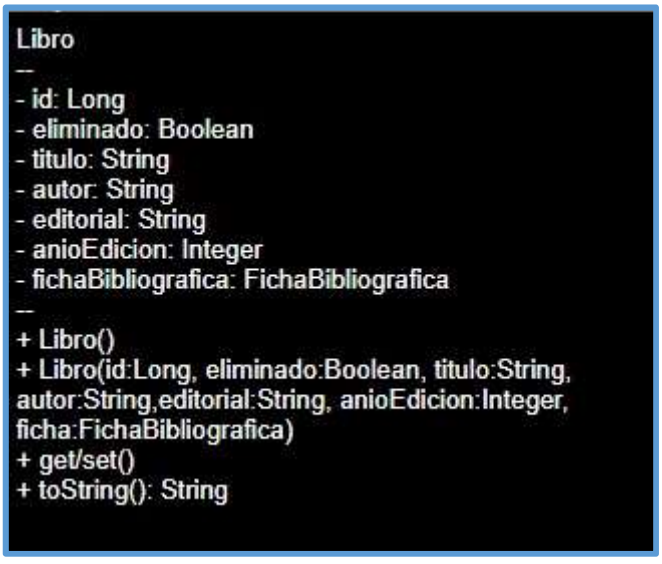
- Lenguaje: Java JDK (JDK 21 o superior)
- Bases de Datos: MySQL
- Persistencia: JDBC
- Arquitectura: Patrón DAO + Capa Service.
- Dominio Elegido: Libro → Fichabibliografica.

El alcance del proyecto incluye la implementación del CRUD completo (Crear, Leer, Actualizar, Eliminar) para ambas entidades y el uso de baja lógica (Soft Delete) para mantener el historial de registros.

Diseño del Proyecto:

2.1: Modelo de Dominio: Entidades

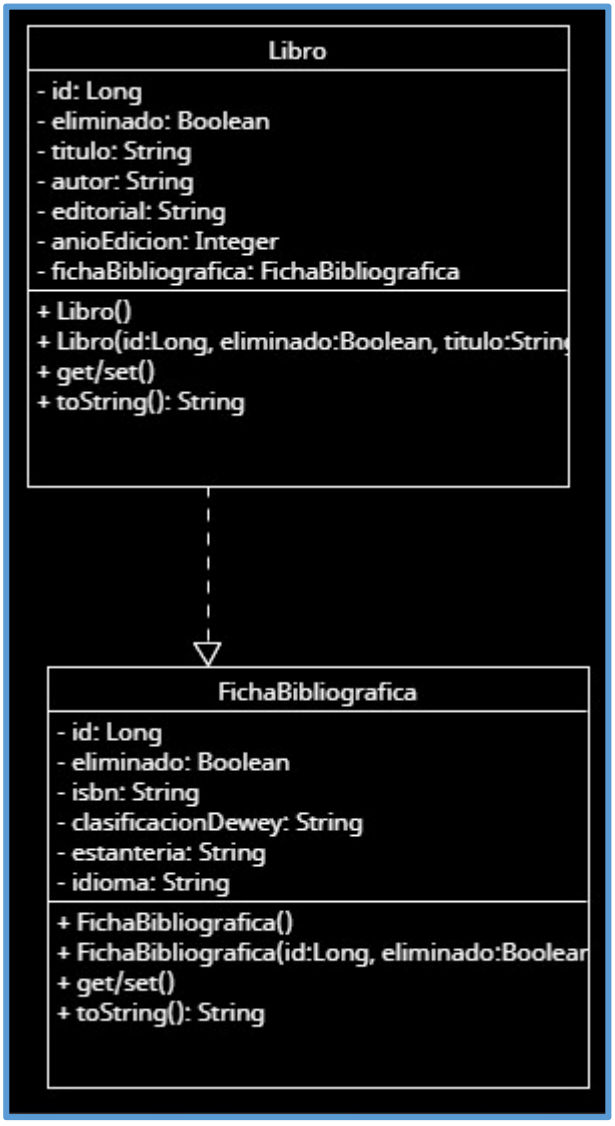
Se eligió el dominio Libro como la Clase A (la que referencia) y FichaBibliografica como la Clase B (la referenciada), con la siguiente estructura.



Clase A: Libro

Clase B: FichaBibliografica

Diagrama UML



2.2: Diseño de Clases (Diagrama UML)

El diagrama de clases representa la arquitectura del proyecto, demostrando la separación de responsabilidades y la aplicación del patrón DAO.

- **Estructuras de Capas:** El diagrama muestra la división de Entidades (Libro, FichaBibliografica), DAO (GenericDao, LibroDao, FichaBibliograficaDao), Service (LibroService, FichaBibliograficaService) y la clase de Configuración (DatabaseConnection).
- **Patrón DAO:** Se define la interfaz "GenericDao<T>" para estandarizar las operaciones CRUD. Las clases concretas "LibroDao", "FichaBibliograficaDao" se encargan de las consultas SQL específicas, recibiendo la java.sql.Connection para participar en transacciones.
- **Relación:** La clase "Libro" contiene la referencia (fichaBibliografica) a la clase FichaBibliografica, estableciendo la asociación 1 → 1 unidireccional.

2.3: Diseño de la Persistencia

La relación 1 a 1 unidireccional se implementó en la base de datos "biblioteca_db" mediante dos tablas (libro y ficha_bibliografica). Para asegurar la cardinalidad 1 a 1 y la integridad referencia, la tabla ficha_bibliografica se enlaza con libro mediante una Clave Foránea Única (UNIQUE FOREING KEY) que apunta al ID del libro.

SQL:

```sql
CREATE TABLE ficha_bibliografica (
    id INT AUTO_INCREMENT PRIMARY KEY,
    isbn VARCHAR(17) UNIQUE,
    clasificacion_dewey VARCHAR(20),
    estanteria VARCHAR(20),
    idioma VARCHAR(30),
    eliminado BOOLEAN NOT NULL DEFAULT FALSE
);
CREATE TABLE libro (
    id INT AUTO_INCREMENT PRIMARY KEY,
    titulo VARCHAR(150) NOT NULL,
    autor VARCHAR(120) NOT NULL,
    editorial VARCHAR(100),
    anio_edicion INT,
    eliminado BOOLEAN NOT NULL DEFAULT FALSE,

    -- Relación 1 a 1
    ficha_id INT UNIQUE,
    CONSTRAINT fk_libro_ficha FOREIGN KEY (ficha_id)
        REFERENCES ficha_bibliografica(id)
        ON DELETE CASCADE
);
```

Validaciones y Reglas del Negocio:

Las reglas del negocio y las validaciones de datos se implementaron en la capa Service para asegurar la calidad de los datos y la integridad de la lógica antes de la persistencia.
Se implementaron dos tipos de validaciones: validación de datos (formatos y obligatoriedad) y validación de lógica de negocio (unicidad y estados).

Validaciones de Integridad de Datos:

Estas validaciones son cruciales en la entrada de datos (Creación y Actualización):

- Campos Obligatorios: (NOT NULL): Se verifica que los campos esenciales no estén vacíos o nulos.
  - Libro: título, autor.
  - FichaBibliografica: isbn
- Restricciones de Longitud: Se asegura que los String no excedan la longitud máxima definida en el modelo de bases de datos

Reglas del Negocio:

- Unicidad del ISBN: El campo “isbn” en “FichaBibliografica” debe ser único. Esta regla se valida tanto en la capa de Java (para dar un mensaje al usuario) como en la base de datos (mediante la restricción UNIQUE en la tabla).
- Unicidad de la Relación 1 a 1: Al crear “Libro”, se verifica que la “FichaBibliografica” que se intenta asociar no haya sido previamente utilizada por otro “Libro” activo. Esto es reforzado por la Clave Foránea Única en la tabla “Libro”.
- Asociación Valida: Para crear un Libro, el objeto FichaBibliografica asociado debe ser una entidad valida y existente en la base de datos.

Reglas de Estado (Baja Lógica):

Soft Delete: Se implementa la regla de negocio de que ningún registro debe ser eliminado físicamente. La operación “eliminar” consiste en establecer el campo “eliminado=TRUE” en la entidad correspondiente libro o fichabibilografica.
Todas las operaciones de lectura, consultas y listados deben incluir la condición “Where eliminado = False” para operar solo sobre registros activos.
Impedir operaciones en inactivos las operaciones de “Actualizar” o “Eliminar” deben fallar si el ID ingresado corresponde a una entidad que ya está marcada como eliminado = TRUE.

Pruebas Realizadas:

```
===== SISTEMA DE BIBLIOTECA =====
1. Crear ficha bibliogr□fica
2. Crear libro
3. Buscar libro por ID
4. Listar libros
5. Actualizar libro
6. Eliminar libro
7. Salir
Seleccione una opci□n:
```

Captura del Menú

```
1. Crear ficha bibliogr�fica
2. Crear libro
3. Buscar libro por ID
4. Listar libros
5. Actualizar libro
6. Eliminar libro
7. Salir
Seleccione una opci�n: 4
Libro{id=1, eliminado=false, titulo='Cien a�os de soledad', autor='Gabriel Garc�a M�rquez', editorial='Sudamericana', anioEdicion=1967, fichaBibliografica=978-84-376-0494-7}
Libro{id=2, eliminado=false, titulo='El Principito', autor='Antoine de Saint-Exup�ry', editorial='Reynal & Hitchcock', anioEdicion=1943, fichaBibliografica=978-0-15-601219-5}
Libro{id=3, eliminado=false, titulo='1984', autor='George Orwell', editorial='Secker & Warburg', anioEdicion=1949, fichaBibliografica=978-0-452-28423-4}
```

Consultas sobre los libros disponibles

Conclusiones:

El proyecto hemos realizado cumpliendo con todos los objetivos solicitados en el Trabajo Integrador Final, logrando la implementación de una aplicación en Java que modela la relación 1 → 1 unidireccional y utiliza el patrón DAO/Service de manera efectiva. El requerimiento de transacciones explicitas con JDBS fue el desafío central, asegurándonos que la persistencia de las dos entidades relacionadas se manejen como una única operación atómica, lo cual es fundamental para el manejo de la integridad de los datos del sistema.

Como mejora para futuras iteraciones del proyecto, podríamos proponer algunos puntos como:

- Configuración Externas: Implementar la lectura real de las credenciales de conexión (URL, USER, PASSWORD) desde un archivo ".properties" para cumplir estrictamente con el requisito de "lecturas de propiedades externas"
- Abstracción de Excepciones: Crear excepciones de dominio personalizadas y no verificadas "PersistenceException" para encapsular las "SQLException" de JDBC, desacoplando la capa Service de la API de bases de Datos.
- Interfaz de Usuario: Desarrollar una interfaz gráfica o una API REST para reemplazar el menú de la consola.