Trabajo práctico integrador

Programación II

Integrantes:

Lucas Martin Monsón
Fernando Agustín Palacios
Agustina Aguilera

Informe

El dominio planteado para el desarrollo de este trabajo es una biblioteca, donde se requiere gestionar libros y sus fichas bibliográficas, permitiendo agregar, eliminar y consultar el contenido de dicha biblioteca.
La entidad A está conformada por los Libros, mientras que la entidad B está conformada por las FichasBibliograficas.
Ambas entidades se relacionan en una asociación 1 a 1, donde cada libro posee una única ficha bibliográfica asociada.
Sin embargo, desde el punto de vista teórico, se considera que el libro no tiene conocimiento directo de su ficha, siendo la FichaBibliografica la que mantiene la referencia al libro. Esta elección conceptual se basa en que la ficha depende del libro para existir, mientras que el libro puede entenderse como la entidad principal e independiente dentro del dominio.

Diseño
El diseño presenta una estructura de código **orientada a objetos** con clases bien definidas en el paquete entities.
Cada entidad tiene su id y un campo eliminado para representar bajas lógicas sin eliminar físicamente los registros de la base.

- **Relación 1→1 (Libro → FichaBibliografica):**

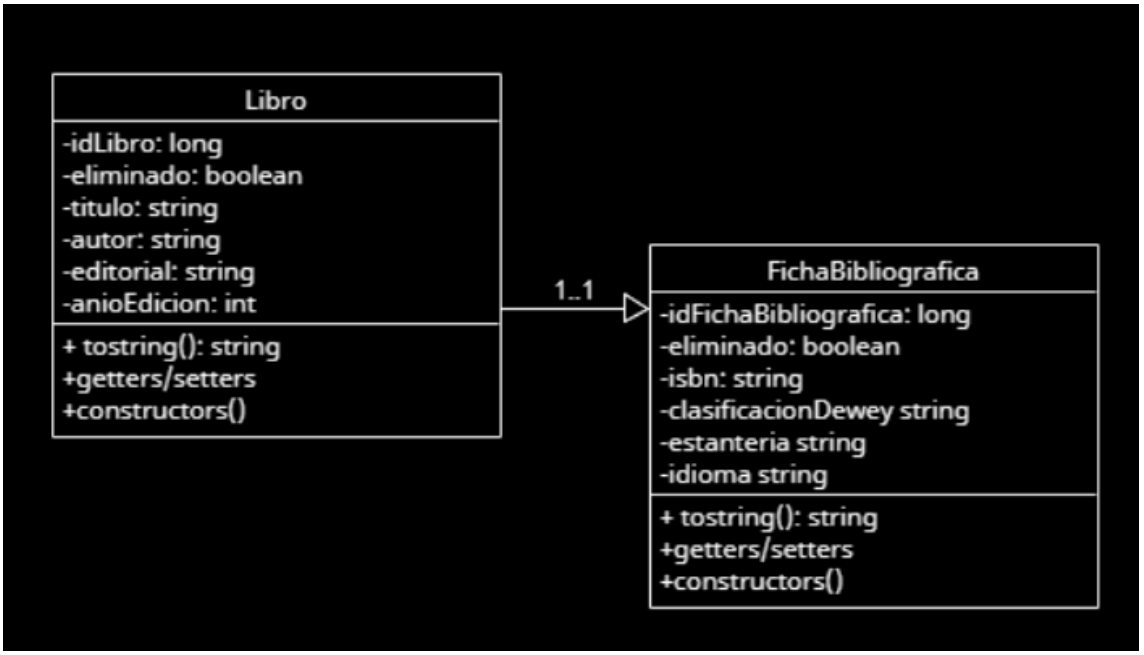
  La clase Libro contiene un atributo privado FichaBibliografica ficha, cumpliendo así la relación unidireccional de 1 a 1.
- **Implementación en base de datos:**

Se decidió usar una clave foránea única en la tabla FichaBibliografica, que apunta a la tabla Libro.
De esta manera, cada ficha queda asociada a un solo libro, pero ambas tablas siguen siendo independientes.
Además, esta opción hace más simple el manejo de altas, bajas y actualizaciones, especialmente cuando se aplican eliminaciones en cascada.

- **Tipos de datos:**

  Los identificadores (idLibro, idFichaBibliografica) son de tipo long y los campos eliminados son booleanos, porque recordemos que no van a ser eliminados del sistema, sino que serán o no visibles.
  Los atributos textuales utilizan strings.

Arquitectura por capas



El proyecto se organiza de la siguiente manera:

- Capa ENTITIES: Contiene las clases que representan las tablas de la base de datos
- Capa DAO (Interfaces): Contiene las interfaces genéricas. Declaran operaciones de persistencia y aíslan a la lógica del código SQL.
- Capa DAO (Implementación): Preparan y ejecutan las sentencias SQL, reciben los datos y manejan las transacciones.
- Capa SERVICIOS: Centralizan las reglas de negocio.
- Capa Main: articula el menú consola que interactúa con el usuario.

Persistencia

Estructura de la base: compuesta por dos tablas: Libro y FichaBibliografica en una relación 1 a 1.

Orden de operaciones

- Insert
    - Insertar Libro y obtener ID
    - Insertar Ficha usando ese ID
- Delete lógico
    - Ambas entidades tienen el atributo eliminado.

Manejo de transacciones

Se hace COMMIT en métodos que afectan múltiples tablas.
Se hace ROLLBACK si falla cualquiera de las operaciones.

Servicios y transacciones

El proyecto usa una interfaz genérica ServicioGenerico<T> que define CRUD básico para cualquier entidad. Sobre esta base:

ServicioLibro:

- Implementa transacciones obligatorias: abre una conexión, hace setAutoCommit(false) y ejecuta operaciones compuestas (crear/actualizar libro + ficha).
- Si todo sale bien hace commit; si ocurre un error, rollback.
- Válida unicidad del ISBN y aplica la relación 1→1 entre Libro y FichaBibliografica (la ficha se crea y actualiza siempre junto al libro).
- Finalmente restablece autoCommit(true) y cierra la conexión.

ServicioFichaBibliografica:

- CRUD simple sin transacciones compuestas.

El CRUD de FichaBibliografica se implementa en su servicio específico y sigue el esquema básico de crear, leer, actualizar y eliminar. Cada operación abre su propia conexión a la base de datos y usa los métodos del DAO para ejecutar la acción solicitada.

La creación inserta una nueva ficha asociada a un libro; la lectura permite obtener una ficha por ID o listar todas; la actualización modifica los datos bibliográficos (ISBN, Dewey, estantería, idioma); y la eliminación quita la ficha correspondiente.

AppMenu (consola)

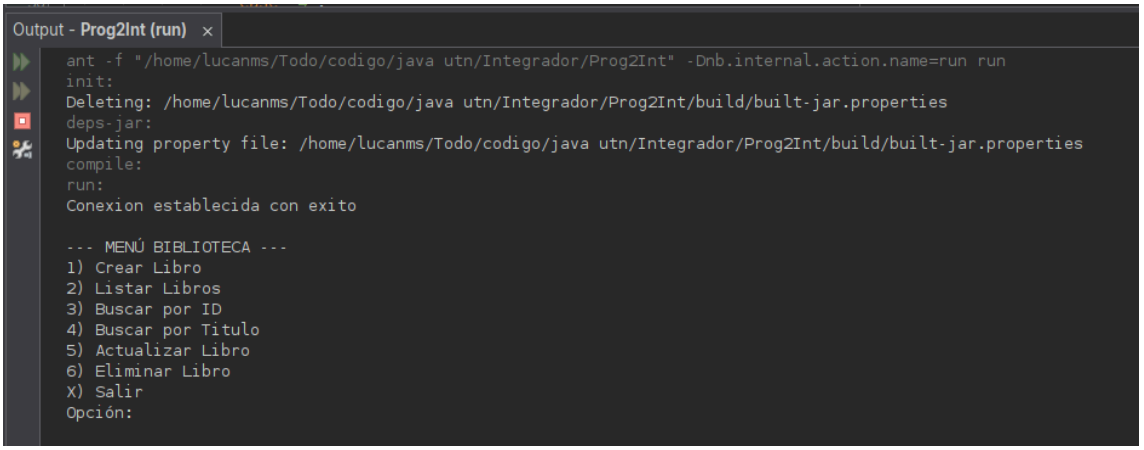
Main llama a AppMenu.iniciar(), que muestra el menú y convierte las opciones a mayúsculas.

La consola permite el CRUD completo de libros: crear, listar, buscar por ID, buscar por título, actualizar y eliminar (lógico).

Incluye:

- Validaciones básicas (campos vacíos, años numéricos, IDs válidos).
- Manejo de errores como ISBN duplicado, IDs inexistentes o fallos de BD.
- Mensajes claros de éxito o error.

Capturas de ejecución y base de datos:



Este trabajo tiene solo tres integrantes porque el cuarto dejó de responder y no participó.