



Trabajo Final Integrador

Alumnos:

- Emilia Gómez Juárez. Mat. 101213
- Emanuel Gómez Juárez.
- Matías Farfán, legajo: 4839.
- Marianela Guerrero, legajo: 1120.

Carrera: Tecnicatura universitaria en programación.

Asignatura: Programación II.

Fecha: 16/11/2025.

Indice

Integrantes y roles.....	3
Elección del dominio y justificación.....	3
Diseño.....	3
Arquitectura por capas.....	3
Persistencia.....	3
Validaciones y reglas de negocio.....	3
Pruebas realizadas.....	3
Conclusiones y mejoras futuras.....	3
Citar fuentes y herramientas utilizadas.....	3

Integrantes y roles

- Emilia Gómez Juárez:
- Emanuel Gómez Juárez:
- Matías Farfán
- Marianela Guerrero

Elección del dominio y justificación

Dominio elegido: Libro → FichaBibliografica

Se seleccionó este dominio debido a que representan elementos del ámbito familiar y permiten la complejidad y profundidad requerida para el desarrollo del trabajo práctico integrador.

Diseño

Para el diseño del trabajo se siguieron determinados requerimientos técnicos base que sirvieron para la toma de decisiones y modelado del desarrollo.

Partimos de la organización de un diagrama UML (adjuntado en el repositorio de github). Este diagrama sirvió de arquitectura base para establecer qué clases necesitábamos, qué atributos y métodos debía contener cada una. Sobre esta base se definieron las claves primarias (PK) de cada entidad y las claves foráneas (FK) que materializan las asociaciones en el modelo relacional.

Se tomó mayor análisis para las relaciones 1 a 1 solicitadas y cómo debía ser su implementación a través de las PK y FK. Estas decisiones de diseño garantizan la consistencia referencial de los datos y reflejan en la base de datos las restricciones del dominio expresadas originalmente en el diagrama UML.

Arquitectura por capas

Se implementaron las siguientes capas:

- Entidades: Esta capa presenta los modelos utilizados para estructurar los datos para el modelo del dominio del desarrollo. Es la estructura lógica del dominio implementado. Encargada de estructurar las clases con sus atributos y

métodos representando las entidades principales requeridas y sirviendo como base para las operaciones que realizan las otras capas.

- DAO: Esta capa es la que se encarga de relacionar el trabajo con el acceso a los datos persistidos, posee el JDBC ejecutado y se encarga del manejo de resultados. Involucrado en la función de persistencia de los datos y simplifica los detalles de persistencia.
- Servicio: En esta capa se establecen las validaciones de los datos y manejo de transacciones, son las encargadas de la coordinación con los DAOs y la correcta implementación de las reglas de negocio y casos de uso especificados de la aplicación. Además, en los casos que es requerido, se encarga del manejo de transacciones asegurando la consistencia de los datos ante errores
- Presentación: Esta es la primera instancia, la responsable de la interacción con el usuario. En esta capa se interpretan los comandos que recibe por parte del usuario y muestra los resultados obtenidos.

Persistencia

La base consiste en dos tablas, Libro y FichaBibliografica. Estas están estructuradas de la siguiente manera.

tabla libro:

- id: es la Primary Key de la tabla, tiene la propiedad AUTO INCREMENT que se encarga del manejo de los id automáticamente
- eliminado: atributo que por defecto es FALSE
- titulo: string no nulo que puede tener como máximo 150 caracteres
- autor: String no nulo que puede tener como máximo 120 caracteres
- editorial: String que puede tener hasta 100 caracteres
- anio_edicion: Número tipo integrer
-

tabla fichaBibliografica:

- id: es la Primary Key de la tabla, tiene la propiedad AUTO INCREMENT que se encarga del manejo de los id automáticamente
- eliminado: atributo que por defecto es FALSE
- isbn: String con propiedad de ser unico de hasta 17 caracteres
- clasificacion_dewey: String que puede tener hasta 20 caracteres
- estanteria: String que puede tener hasta 20 caracteres

- idioma: String que puede tener hasta 30 caracteres
- libro_id: FK, referencia a la PK de la tabla libro

Orden de operaciones y transacciones (commit/rollback)

El orden de las operaciones consiste en los siguientes pasos:

- 1) orden del usuario de ejecutar una operación. Ej: Insert
- 2) conexión a base de datos en service
- 3) validación del input
- 4) Ejecución del comando en caso de validación exitosa. Ej: se inserta valor en tabla de base de datos
- 5) Validación de resultado del insert. Ej: creación exitosa de libro
- 6) Creación de asociación con otra tabla y coherencia de datos
- 7) Commit de la transacción
- 8) devolución de respuesta. Ej: return de libro creado

Las transacciones ocurren cuando se busca ejecutar el CRUD sobre la base de datos.

Para las mismas se implementa:

- El try encargado de establecer la conexión y realizar el commit en caso de éxito.
- El catch encargado de ejecutar el rollback en caso de error
- El finally encargado de siempre cerrar la conexión

Validaciones y reglas de negocio

Se realizan varias validaciones a lo largo del código que permiten el correcto manejo de los datos y funcionalidad del sistema. Podemos categorizarlas de la siguiente manera:

- Validaciones de entrada:

Todos los valores que ingresan son validados para asegurar que no sean valores nulos y que sean los apropiados para el tipo de orden que se pretende ejecutar garantizando de esa manera la calidad e integridad de los datos. Entre estas podemos incluir las verificaciones de los tipos de datos como numéricos o de Strings, la valoración de inputs nulos u obligatorios, validación de fechas, tanto por el formato requerido como para la lógica seteada.

- Validaciones de dominio:

En esta incluimos las validaciones sobre la integridad y coherencia de los datos. Por ejemplo, que no haya dos referencias a un mismo atributo. Por ejemplo, que dos libros no hagan referencia a la misma ficha bibliográfica asegurando la integridad de las relaciones entre entidades y restricciones establecidas.

Pruebas realizadas

(capturas del menú y consultas SQL útiles).

Conclusiones y mejoras futuras

Con el trabajo realizado se logró implementar de manera integral las diferentes capas del desarrollo y vincularlas a una base de datos relacional. La dinámica de trabajo en equipo resultó clave para distribuir tareas, resolver problemas de forma colaborativa y enfrentar escenarios similares a los de un entorno profesional. Esta experiencia práctica contribuyó significativamente a la fijación de conocimientos, al ofrecer un contexto realista en el que aplicar los conceptos teóricos trabajados durante la cursada de la materia de Programacion II.

A partir de las lecciones aprendidas durante el desarrollo del trabajo se identificaron áreas de mejoras futuras entre las que podemos destacar la implementación de validaciones más avanzadas, integración con un sistema de logueo que permita registrar y rastrear errores que pueda presentar el código y auditoría de los datos, automatización de las pruebas para una detección temprana de fallos y mantenibilidad del sistema.

Citar fuentes y herramientas utilizadas