

Informe corto — Fase 3 (Implementación C++)

Objetivo: implementar el modelo de la Fase 2 (3FN) en C++ usando structs, vectores, menú CRUD y persistencia en archivos .txt.

Lo aprendido 1) La normalización (1FN→3FN) simplificó el código porque evitó duplicar datos: por ejemplo, los datos del Autor no se repiten en Libro; solo se guarda el id_autor y se “hace join” al imprimir. 2) Las relaciones se manejan con validaciones en C++: antes de registrar un préstamo, el programa verifica que existan el Libro y el Estudiante, y que el libro no tenga un préstamo activo. 3) La persistencia con fstream (CSV simple) es suficiente para CRUD básico; sin embargo, muestra limitaciones (por ejemplo, comas en los textos), lo cual motiva el paso posterior a un motor real (SQLite). 4) El uso de std::vector es adecuado para listas pequeñas/medianas; para escalabilidad se recomienda índices (map) o una BD.

Decisiones de diseño • Estructuras de datos: structs simples (Autor, Libro, Estudiante, Prestamo) y una clase BibliotecaDB como “repositorio” en memoria. • Validaciones clave incluidas: id duplicado, existencia de FKs, disponibilidad de libro para nuevo préstamo, y devolución. • Formato de fechas libre (yyyy-mm-dd) sin validación estricta para mantener foco en la lógica de BD.

Siguientes pasos • Migrar a SQLite y reemplazar los bucles por consultas SQL reales con claves foráneas y restricciones únicas. • Mejorar el parser CSV o usar una librería (cuando sea permitido) para soportar comillas/escapes. • Agregar pruebas unitarias para automatizar la validación de los casos de error.