

## Aprendizajes obtenidos del Reto 3 – Diseño de Base de datos.

Durante el desarrollo de este reto aprendí a conectar la teoría de bases de datos con una implementación práctica en C++. Las entregas se organizaron en tres fases a lo largo de una semana y media, lo que me permitió llevar un control claro de los avances y trabajar en un proyecto desarrollado de forma progresiva. Esta división en etapas fue muy valiosa, ya que me ayudó a comprender que un sistema complejo no se construye de golpe, sino paso a paso, validando cada componente antes de continuar.

En la Fase 1 – Diseño conceptual, elaboré el diagrama entidad-relación. Esta experiencia me permitió aprender de forma práctica a identificar entidades, atributos, claves primarias y foráneas, así como las cardinalidades entre tablas. Al representarlas gráficamente comprendí que un buen diseño inicial evita problemas posteriores en la implementación. Un ejemplo concreto fue entender que un autor puede tener varios libros (1:N) y que cada préstamo conecta un libro con un estudiante. Reflexioné también sobre la importancia de definir restricciones desde el principio, como evitar que un libro sin autor registrado exista en el sistema o que un préstamo se genere sin estudiante asociado.

La Fase 2 – Normalización fue un reto completamente nuevo para mí. En ella traduje el modelo conceptual al modelo relacional y lo normalicé hasta la Tercera Forma Normal (3FN). Descubrí que la normalización no es solo una teoría abstracta, sino una herramienta concreta para evitar redundancias y mantener la integridad de los datos. Un aprendizaje clave fue que no es necesario repetir el nombre del autor en cada libro, basta con almacenar el ID\_Autor. Esto garantiza que, si un autor cambia de nombre o se corrige un error tipográfico, el ajuste se hace una sola vez y se refleja automáticamente en todos sus libros. Comprendí que esta práctica también previene anomalías en inserciones y eliminaciones, lo que mejora la consistencia del sistema.

La Fase 3 – Implementación en C++ resultó ser la más desafiante y enriquecedora, ya que implicó convertir el modelo en código real. Representé las tablas mediante structs y usé vectores para almacenar los registros en memoria. Programé un menú interactivo con operaciones CRUD, lo que me permitió comprobar en la práctica cómo un diseño abstracto se transforma en un sistema funcional. También implementé la persistencia de datos con archivos de texto en formato CSV, lo cual fue un aprendizaje clave porque me acercó a la manera en que los sistemas almacenan información de manera simple pero efectiva. Otro punto importante fue el manejo de errores: aprendí a validar IDs duplicados, a evitar que se preste un libro ya ocupado y a gestionar la devolución como un campo opcional. Estos detalles, aunque pequeños, marcaron la diferencia en la calidad del programa.

En conclusión, este proyecto me enseñó a pasar de un diseño conceptual a una solución práctica desarrollada en código. Comprendí lo útil que es la validación de datos y el valor de versionar el trabajo en GitHub, ya que me permitió organizar entregas, corregir errores y registrar la evolución del proyecto. El reto reforzó y amplió mis conocimientos de bases de datos y programación en C++, mostrándome que una buena planificación facilita la implementación exitosa de un sistema funcional. Además, me dejó la motivación de seguir aprendiendo cómo mejorar un prototipo básico hacia un sistema más robusto, incorporando nuevas funcionalidades y mejores prácticas de desarrollo.