# Proyecto Estructura y Base de Datos

Scarpato Genaro, Smilari Lucas, Matías Seba Mallo

*Universidad Tecnológica nacional – Instituto Nacional Superior del Profesorado Técnico, Buenos Aires, Argentina*

## **Alcance del Proyecto**

Un usuario nos plantea la necesidad de informatizar la biblioteca pública llamada “**Biblioteca MLG**”, con el fin de optimizar la renta de libros. Los socios de la biblioteca pueden adquirir libros, por tiempo plazos determinados y últimamente el registro, que se lleva en papel, a resultado poco práctico.

## **Problemática para resolver**

Se nos pide resolver la optimización en los procesos de renta de libros, así como, su seguimiento y control, para un mejor servicio.

i. Digitalizar los documentos y registros ya existentes en forma analógica.

ii. Optimizar y agilizar los procesos de renta.

iii. Llevar un control más riguroso de cada uno de los clientes.

Para cumplir con estas necesidades, se estimas que los datos más relevantes a informatizar son los que componen toda la identidad de los **Libro**, así también, como los de cada **Socio**.

## **Relevamiento de las tareas**

Lo primero a realizar, es una división y caracterización de datos para dar constancia del volumen de estos y la cantidad de tiempo que conllevaría poder digitalizarlos.

Posteriormente, hacer un esquema con las fechas de trabajo, así también, como sus respectivas muestras y correcciones con el equipo y el cliente.

## **Diagramas Realizados**

Para completar las bases de datos determinamos las siguientes tablas con sus respectivos campos, se muestra en **Negrita** aquellos datos, clave primaria:

Tabla Libro: **ID\_libro**, ISBN, Titulo, ID\_autor, ID\_genero, ID\_editorial, Stock.

Tabal Autor: **ID\_autor**, Nombre, Apellido.

Tabla Editorial: **ID\_editorial**, Nombre.

Tabal Genero: **ID\_genero**, Tipo.

Tabal Alquiler: **ID\_alquiler**, ID\_libro, ID\_socio, Fecha\_adquirido, Fecha\_entregado.

Tabla Socio: **Código de socio**, **DNI**, nombre completo, teléfono, mail, dirección.

A continuación, se muestra el resultado planteado en Excel de lo que se acaba de exponer.

**Tabla Libro**

****

**Tabal Autor Tabla Editorial**

**** ****

**Tabla Socio**

****

**Tabal Alquiler Tabal Genero**

**** ****

El quipo determino que las tablas quedaran de tal forma, gracias a que previamente, de realizó el **Diagrama Entidad Relación**.

Diagrama

Descripción generada automáticamente

Posterior a la realización de dicho diagrama y el planeamiento de las tablas en Excel, proseguimos mostrando el modelo relacional, utilizando como gestor phpmyadmin, vemos las tablas ya armadas, hecho con la herramienta “diseñador” del propio gestor:

Interfaz de usuario gráfica, Diagrama

Descripción generada automáticamente

## **Diccionario de Datos**

**Una captura de pantalla de una red social

Descripción generada automáticamente**

**Imagen que contiene Interfaz de usuario gráfica

Descripción generada automáticamente**

**Aplicación

Descripción generada automáticamente con confianza media**

**Aplicación

Descripción generada automáticamente con confianza media** **Una captura de pantalla de una red social

Descripción generada automáticamente** Una captura de pantalla de una red social

Descripción generada automáticamente

## **Consultas:**

Mostramos algunas consultas que pueden ser de utilidad para la implementación con el programa terminado.

**Catalogo**

Esta consulta nos permite tener una lista, muy útil, para utilizar en caso de que se desee buscar o mostrar todos los libros que hay en la biblioteca, junto con algunos datos de suma importancia, tales como el Stock, el Título, el Autor, el Género y la Editorial.

SELECT Stock, Titulo, CONCAT(a.Nombre, ' ', a.Apellido) AS Autor, Tipo Genero, e.Nombre Editorial FROM libro l

INNER JOIN autor a ON a.ID\_autor = l.ID\_autor

INNER JOIN genero g ON g.ID\_genero = l.ID\_genero

INNER JOIN editorial e ON e.ID\_editorial = l.ID\_editorial;

Resultado de la consulta:

Tabla

Descripción generada automáticamente

**Vigencia**

A continuación, mostramos 2 consultas con sus respectivos códigos y resultados. Esto se debe a que estamos tratando la misma consulta con la diferencia de una simple condición, vemos una gran utilidad en ambas. Encontramos en la primera a los Socios, sus Contactos, el nombre del Libro, el Plazo que se presto el libro y cuánto tiempo en días quedan para devolver el libro.

SELECT CONCAT(s.Nombre, ' ', s.Apellido) Socio, CONCAT('TEL: ', s.Telefono, ' MAIL: ', s.mail) Contacto, l.titulo Libro, DATEDIFF(Fecha\_entregado,Fecha\_adquirido) AS Plazo, DATEDIFF(CURDATE(),Fecha\_adquirido) AS Vigencia

FROM alquiler a

INNER JOIN socio s ON s.ID\_socio = a.ID\_socio

INNER JOIN libro l ON l.ID\_libro = a.ID\_libro;

Resultado de la consulta:

Texto

Descripción generada automáticamente con confianza baja

En esta otra la única diferencia es, WHERE DATEDIFF(CURDATE(), Fecha\_adquirido) <= 0, y nos muestra el registro antes mencionado, de cada socio para el cual se este fuera de vigencia y deba contactárselo.

SELECT CONCAT(s.Nombre, ' ', s.Apellido) Socio, CONCAT('TEL: ', s.Telefono, ' MAIL: ', s.mail) Contacto, l.titulo Libro, DATEDIFF(Fecha\_entregado,Fecha\_adquirido) AS Plazo, DATEDIFF(CURDATE(),Fecha\_adquirido) AS Vigencia

FROM alquiler a

INNER JOIN socio s ON s.ID\_socio = a.ID\_socio

INNER JOIN libro l ON l.ID\_libro = a.ID\_libro

WHERE DATEDIFF(CURDATE(), Fecha\_adquirido) <= 0;

Resultado de la consulta:



**Cantidad por Genero**

Esta consulta resulta útil para mostrar una estadística de aquellos géneros que más posee la biblioteca.

Resultado de la consulta:

Tabla

Descripción generada automáticamenteSELECT g.Tipo Genero, COUNT(l.ID\_genero) Cantidad

FROM libro l

INNER JOIN genero g ON l.ID\_genero = g.ID\_genero

GROUP BY l.ID\_genero;

**Libros más leídos**

Nos muestra la estadística de los libros que son más leídos dentro de la Biblioteca.

Resultado de la consulta:

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamenteSELECT l.Titulo Libro, COUNT(a.ID\_libro) Solicitado FROM alquiler a

INNER JOIN libro l ON a.ID\_libro = l.ID\_libro

GROUP BY a.ID\_libro

ORDER BY COUNT(a.ID\_libro) DESC;

**Lectores estrella**

Ante la posibilidad de que la biblioteca quiera reconocer a aquellos lectores que son mas asiduos y recurrentes, esta consulta resultaría altamente efectiva.

Resultado de la consulta:

Imagen que contiene Texto

Descripción generada automáticamenteSELECT CONCAT(s.Nombre, ' ', s.Apellido) NombreCompleto, COUNT(a.ID\_alquiler) Lecturas FROM alquiler a

INNER JOIN socio s ON s.ID\_socio = a.ID\_socio

GROUP BY CONCAT(s.Nombre, ' ', s.Apellido)

ORDER BY COUNT(a.ID\_alquiler) DESC;

## **Recursos Utilizados:**

Stack Informático:

• Para el programa y la realización de los algoritmos se utilizó el entorno **Clion** y toda la codificación está hecha en el lenguaje de programación C, adjuntamos página oficial.

https://www.jetbrains.com/es-es/clion/

• Para el armado de los diagramas entidad relación se utilizó **darw.io**.

https://www.drawio.com/

• En cuanto al modelo relacional, el armado de la base de datos y la administración de consultas SQL utilizamos el gestor web **phpmyadmin**.

https://www.phpmyadmin.net/