



ESCUELA POLITÉCNICA NACIONAL

ESCUELA DE FORMACIÓN DE TECNÓLOGOS



BASE DE DATOS

ASIGNATURA:

BASE DE DATOS

PROFESOR:

Ing. Sergio Granizo MSc.

PERÍODO ACADÉMICO:

2025-B

PROYECTO GRUPAL (FIN DEL SEMESTRE)

SISTEMA DE GESTIÓN DE BIBLIOTECAS

PROYECTO SEGUNDO BIMESTRE



ESTUDIANTES
(3 integrantes):

Ariel Paz y Carlos Serrano

DEFINICIÓN DEL PROBLEMA

Un sistema de una biblioteca puede presentar problemas en cuanto a gestión manual y desorganizada de la información que maneja, donde el control de usuarios, libros, ejemplares y préstamos suele realizarse de manera física en cuadernos, hojas sueltas o archivos sin seguridad ni integridad.

La base de datos sistema_biblioteca presenta soluciones a estos problemas planteando una base de datos desarrollada en el gestor MySQL y que cuenta con funciones, índices, vistas que ayudan al manejo óptimo de la información manejada en una biblioteca.

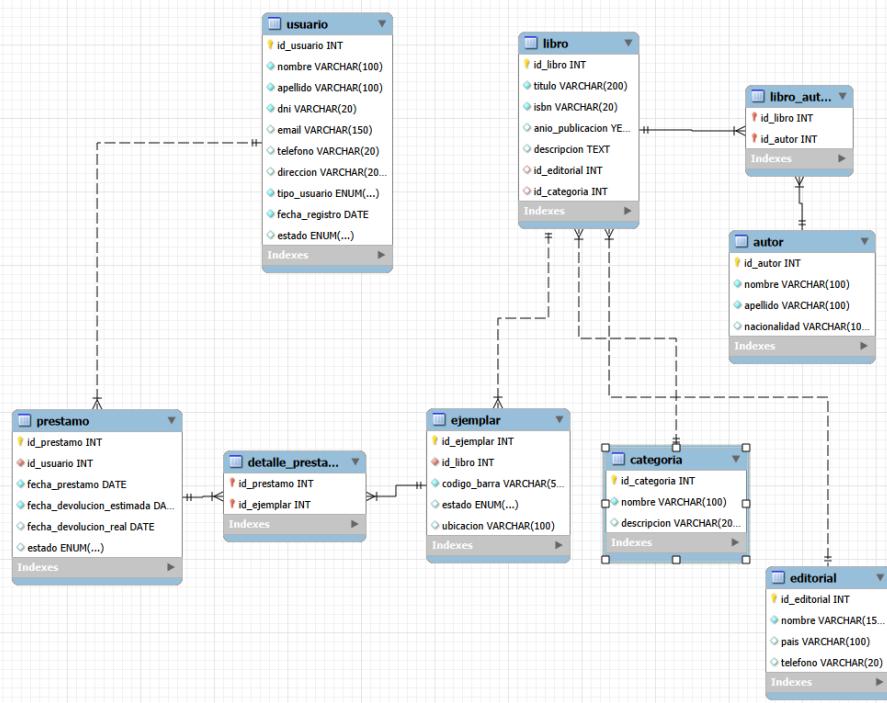
Objetivo general:

Diseñar, implementar y administrar una base de datos relacional utilizando el sistema gestor estándar MySQL, aplicando principios de modelado, normalización y seguridad, con el fin de resolver las problemáticas que puede presentar un sistema de biblioteca que simule necesidades reales de almacenamiento y gestión de información.

Objetivos específicos:

- **Analizar** los requerimientos del caso de estudio para identificar entidades, atributos y relaciones relevantes.
- **Construir** un modelo entidad-relación y transformarlo en un esquema lógico normalizado hasta la tercera forma normal.
- **Implementar** la base de datos en un sistema gestor (MySQL, PostgreSQL, SQL Server, etc.), definiendo tablas, claves primarias y foráneas, así como restricciones de integridad.
- **Desarrollar** consultas SQL que permitan la recuperación, inserción, actualización y eliminación de datos, incluyendo operaciones avanzadas como subconsultas y vistas.
- **Configurar** usuarios, roles y permisos para garantizar la seguridad y correcta administración de la base de datos.
- **Documentar** el proceso de diseño e implementación, incluyendo diagramas, código SQL y ejemplos de resultados obtenidos.

ANÁLISIS Y MODELADO DE DATOS



- La entidad Usuario se separa para centralizar la información de las personas y evitar repetir datos personales en cada préstamo.
- Libro y Ejemplar se diferencian porque un mismo libro puede tener múltiples copias físicas con estados y ubicaciones distintas.
- Autor se separa y se relaciona mediante Libro_Autor para resolver la relación muchos a muchos entre libros y autores sin redundancia.
- Editorial y Categoría se manejan como entidades independientes para mantener la consistencia de datos y facilitar su reutilización en varios libros.
- Préstamo y Detalle_Préstamo se dividen para permitir registrar varios ejemplares en un solo préstamo y mantener la normalización del sistema.

IMPLEMENTACIÓN DE SGBD

```
1 *  CREATE DATABASE Sistema_Biblioteca;
2 *  use Sistema_Biblioteca;
3 *
4 *  CREATE TABLE Usuario (
5     id_usuario INT AUTO_INCREMENT PRIMARY KEY,
6     nombre VARCHAR(100) NOT NULL,
7     apellido VARCHAR(100) NOT NULL,
8     dni VARCHAR(20) UNIQUE NOT NULL,
9     email VARCHAR(150) UNIQUE,
10    telefono VARCHAR(20),
11    direccion VARCHAR(200),
12    tipo_usuario ENUM('Estudiante','Docente','Externo') NOT NULL,
13    fecha_registro DATE NOT NULL,
14    estado ENUM('Activo','Inactivo') DEFAULT 'Activo'
15 );
16
17 *  CREATE TABLE Editorial (
18     id_editorial INT AUTO_INCREMENT PRIMARY KEY,
19     nombre VARCHAR(150) NOT NULL,
20     pais VARCHAR(100),
21     telefono VARCHAR(20)
22 );
23
24 *  CREATE TABLE Categoria (
25     id_categoria INT AUTO_INCREMENT PRIMARY KEY,
26     nombre VARCHAR(100) NOT NULL,
27     descripcion VARCHAR(200)
28 );
29 *  CREATE TABLE Libro (
30     id_libro INT AUTO_INCREMENT PRIMARY KEY,
31     titulo VARCHAR(200) NOT NULL,
32     isbn VARCHAR(20) UNIQUE NOT NULL,
33     anio_publicacion YEAR,
34     descripcion TEXT,
35     id_editorial INT,
36     id_categoria INT,
37     FOREIGN KEY (id_editorial) REFERENCES Editorial(id_editorial),
38     FOREIGN KEY (id_categoria) REFERENCES Categoria(id_categoria)
39 );
40
41
42 *  CREATE TABLE Autor (
43     id_autor INT AUTO_INCREMENT PRIMARY KEY,
44     nombre VARCHAR(100) NOT NULL,
45     apellido VARCHAR(100) NOT NULL,
46     nacionalidad VARCHAR(100)
47 );
48
49 *  CREATE TABLE Libro_Autor (
50     id_libro INT NOT NULL,
51     id_autor INT NOT NULL,
52     PRIMARY KEY (id_libro, id_autor),
53     FOREIGN KEY (id_libro) REFERENCES Libro(id_libro)
54         ON DELETE CASCADE,
55     FOREIGN KEY (id_autor) REFERENCES Autor(id_autor)
56         ON DELETE CASCADE
57 );
58
59 *  CREATE TABLE Ejemplar (
60     id_ejemplar INT AUTO_INCREMENT PRIMARY KEY,
61     id_libro INT NOT NULL,
62     codigo_barra VARCHAR(50) UNIQUE NOT NULL,
63     estado ENUM('Disponible','Prestado','Dañado','Perdido') DEFAULT 'Disponible',
64     ubicacion VARCHAR(100),
65     FOREIGN KEY (id_libro) REFERENCES Libro(id_libro)
66 );
```

```
68 • CREATE TABLE Prestamo (
69     id_prestamo INT AUTO_INCREMENT PRIMARY KEY,
70     id_usuario INT NOT NULL,
71     fecha_prestamo DATE NOT NULL,
72     fecha_devolucion_estimada DATE NOT NULL,
73     fecha_devolucion_real DATE,
74     estado ENUM('Activo','Devuelto','Retrasado') DEFAULT 'Activo',
75     FOREIGN KEY (id_usuario) REFERENCES Usuario(id_usuario)
76 );
77
78 • CREATE TABLE Detalle_Prestamo (
79     id_prestamo INT NOT NULL,
80     id_ejemplar INT NOT NULL,
81     PRIMARY KEY (id_prestamo, id_ejemplar),
82     FOREIGN KEY (id_prestamo) REFERENCES Prestamo(id_prestamo)
83         ON DELETE CASCADE,
84     FOREIGN KEY (id_ejemplar) REFERENCES Ejemplar(id_ejemplar)
85 );
```

Consultas y resultados

```
###Libros actualmente prestados con datos del usuario

SELECT
    u.nombre,
    u.apellido,
    l.titulo,
    e.codigo_barra,
    p.fecha_prestamo,
    p.fecha_devolucion_estimada
FROM Prestamo p
JOIN Usuario u ON p.id_usuario = u.id_usuario
JOIN Detalle_Prestamo dp ON p.id_prestamo = dp.id_prestamo
JOIN Ejemplar e ON dp.id_ejemplar = e.id_ejemplar
JOIN Libro l ON e.id_libro = l.id_libro
WHERE p.estado = 'Activo';
```

	nombre	apellido	titulo	codigo_barra	fecha_prestamo	fecha_devolucion_estimada
▶	Juan	Pérez	Introducción a Bases de Datos	BD-001	2024-03-01	2024-03-10

```
### Usuarios con préstamos retrasados

SELECT
    u.nombre,
    u.apellido,
    u.dni,
    p.fecha_devolucion_estimada,
    DATEDIFF(CURDATE(), p.fecha_devolucion_estimada) AS dias_retraso
FROM Prestamo p
JOIN Usuario u ON p.id_usuario = u.id_usuario
WHERE p.estado = 'Activo'
AND p.fecha_devolucion_estimada < CURDATE();
```

	nombre	apellido	dni	fecha_devolucion_estimada	dias_retraso
▶	Juan	Pérez	12345678	2024-03-10	690

```
### Libros más prestados
```

```
SELECT
    l.titulo,
    COUNT(dp.id_ejemplar) AS veces_prestado
FROM Detalle_Prestamo dp
JOIN Ejemplar e ON dp.id_ejemplar = e.id_ejemplar
JOIN Libro l ON e.id_libro = l.id_libro
GROUP BY l.id_libro
ORDER BY veces_prestado DESC
LIMIT 5;
```

	titulo	veces_prestado
▶	Introducción a Bases de Datos	1
	Cien Años de Soledad	1

```
### Disponibilidad de libros por título
```

```
SELECT
    l.titulo,
    SUM(CASE WHEN e.estado = 'Disponible' THEN 1 ELSE 0 END) AS disponibles,
    COUNT(e.id_ejemplar) AS total_ejemplares
FROM Libro l
JOIN Ejemplar e ON l.id_libro = e.id_libro
GROUP BY l.id_libro;
```

	titulo	disponibles	total_ejemplares
▶	Introducción a Bases de Datos	1	1
	Cien Años de Soledad	1	1

```
###Historial de préstamos de un usuario

SELECT
    u.nombre,
    u.apellido,
    l.titulo,
    p.fecha_prestamo,
    p.fecha_devolucion_real,
    p.estado
FROM Usuario u
JOIN Prestamo p ON u.id_usuario = p.id_usuario
JOIN Detalle_Prestamo dp ON p.id_prestamo = dp.id_prestamo
JOIN Ejemplar e ON dp.id_ejemplar = e.id_ejemplar
JOIN Libro l ON e.id_libro = l.id_libro
WHERE u.id_usuario = 1;
```

	nombre	apellido	título	fecha_prestamo	fecha_devolucion_real	estado
▶	Juan	Pérez	Introducción a Bases de Datos	2024-03-01	NULL	Activo

CONCLUSIONES

1. Se logró construir un esquema entidad-relación y, como tal, una base de datos que pueda abarcar todos los campos que necesita un sistema de bibliotecas para operar correctamente todas las consultas que se necesiten realizar.
2. Se configuraron usuarios con roles y sus respectivos permisos para garantizar la seguridad dentro de la base de datos.
3. Se pudo determinar todas las necesidades que puede presentar un sistema de gestión de una biblioteca y entender lo limitado que es este proceso sin un sistema de base de datos estructurado de manera óptima y precisa.
4. El sistema proporciona una solución robusta y eficiente para la administración moderna de información

