

# SISTEMA DE GESTIÓN DE LIBROS ELECTRÓNICOS

PROGRAMACION ORIENTADA A OBJETOS 1-SIN-  
3A

Nombre del proyecto: *Sistema de Gestión de Libros  
Electrónicos*  
Fecha: 15/11/2025

ESTUDIANTE: CUBERO GUERRERO LENIN WLADIMIR  
[lecularogu@uide.edu.ec](mailto:lecularogu@uide.edu.ec)

## **Etapa 1. Planeación del software**

### **Sistema de Gestión de Libros Electrónicos**

Para el desarrollo de este proyecto se selecciona el **Sistema de Gestión de Libros Electrónicos**, ya que el sistema de libros permite concentrarse en funcionalidades de gestión (alta, baja, modificación, consulta) lo realizare sin la opción de compra de los libros solo por consulta y lectura en línea.

Este sistema está pensado como una aplicación donde los usuarios pueden **consultar un catálogo de libros electrónicos**, mientras que un **administrador** gestiona tanto el catálogo como los usuarios de la plataforma.

### **Objetivo del sistema y usuarios**

#### **2.1. Objetivo general**

Diseñar y planificar un **Sistema de Gestión de Libros Electrónicos** que permita administrar un catálogo de libros y el acceso de usuarios, de manera sencilla, organizada y segura, utilizando conceptos de **programación funcional**.

#### **2.2. Objetivos específicos**

- Permitir que un **administrador** pueda gestionar el catálogo de libros electrónicos.
- Permitir que un **administrador** pueda gestionar los usuarios y sus permisos.
- Permitir que un **usuario consumidor** consulte los libros disponibles sin modificar los datos.
- Definir la estructura de módulos, paquetes y componentes del sistema, de acuerdo con los temas de la unidad.
- Identificar bibliotecas estándar y paquetes externos necesarios para la implementación.

#### **2.3. Tipos de usuarios**

##### **1. Administrador**

- Agregar libros al sistema.
- Modificar la información de los libros.
- Eliminar libros del catálogo.
- Crear usuarios.
- Modificar usuarios (activar, desactivar, cambiar rol).
- Eliminar usuarios.

##### **2. Usuario consumidor**

- Iniciar sesión en el sistema.
- Ver el listado de libros disponibles.
- Buscar libros por título, autor o categoría.

- Ver el detalle de un libro.
- Marcar libros como “favoritos” o “por leer”.

### **3. Alcance y módulos del sistema**

#### **3.1. Alcance del proyecto**

El sistema se enfocará en:

- Gestión interna del catálogo de libros electrónicos.
- Gestión básica de usuarios (administrador y consumidor).
- Funcionalidades de búsqueda y consulta de libros.
- No se desarrollarán funcionalidades de compra, suscripción ni pasarelas de pago.(hasta nuevas actualizaciones)

#### **3.2. Módulos principales**

##### **Módulo 1: Gestión de Autenticación y Usuarios**

- Registro de usuarios (solo por administrador).
- Inicio de sesión.
- Asignación de rol: Administrador / Usuario consumidor.
- Modificación de datos de usuario.
- Activación / desactivación de usuarios.
- Eliminación de usuarios.

##### **Módulo 2: Gestión de Libros**

- Crear libro (título, autor, año, categoría, descripción, ruta de archivo, etc.).
- Editar información de un libro.
- Eliminar libro del catálogo.
- Listar todos los libros.

Disponible solo para el **administrador**.

##### **Módulo 3: Catálogo y Consulta (para el usuario consumidor)**

- Listado público de libros (según permisos).
- Búsqueda de libros por título, autor o categoría.
- Filtro por categoría.
- Visualización del detalle del libro (descripción, autor, año, etc.).

##### **Módulo 4: Favoritos / Lista personal**

- Permitir que el usuario consumidor marque libros como “favoritos” o “por leer”.
- Listar los libros favoritos del usuario.

## **4. Estructura de gestión del sistema (lógica)**

De forma lógica, el sistema se puede organizar en **tres capas**:

### **1. Capa de presentación (interfaz)**

- Consola, interfaz web sencilla o API.
- Muestra menús, listados de libros, formularios de login.

### **2. Capa de lógica de negocio**

- Funciones responsables de:
  - Validar datos de libros y usuarios.
  - Aplicar reglas de negocio (por ejemplo, solo el administrador puede crear libros).
  - Filtrar y ordenar libros para la búsqueda.

### **3. Capa de datos**

- Almacenamiento de libros y usuarios.
- Puede iniciar como una estructura en memoria (slices de structs) o un archivo JSON y luego escalar a base de datos.

## **5. Paquetes a incorporar (en Go) y su uso**

Asumiendo que el desarrollo se realizará en **Golang**, se propone el uso de los siguientes paquetes:

### **5.1. Paquetes de la biblioteca estándar**

- fmt  
Para impresión de mensajes en consola, depuración y salidas simples.
- os  
Para manejo básico de archivos y lectura de variables de entorno (si se requiere).
- encoding/json  
Para serializar y deserializar
- json  
Para almacenar y leer libros y usuarios en formato JSON (útil si se usa archivo como almacenamiento).
- net/http  
En caso de implementar una versión web, permite crear un servidor HTTP para exponer endpoints de consulta y administración.
- time  
Para registrar fechas de creación/modificación de libros o usuarios.
- log  
Para registrar mensajes de error o eventos importantes del sistema.

### **5.2. Paquetes de terceros (opcional)**

- github.com/gorilla/mux  
Descripción de uso:

- Router HTTP más flexible que el router estándar.
- Permite definir rutas como /libros, /libros/{id}, /usuarios/{id}.
- Facilita la separación de controladores (handlers) por módulo (libros, usuarios, etc.).

## 6. Enfoque de programación funcional

Con base en la indicación entregadas, el software se desarrollará **mediante programación funcional** en la medida de lo posible dentro de Go, aplicando:

- **Funciones puras** para la lógica de negocio: Ejemplo: una función que recibe una lista de libros y un término de búsqueda, y devuelve una nueva lista filtrada sin modificar la original.
- **Uso de funciones de orden superior**: Por ejemplo, funciones que reciben como parámetro otra función para aplicar filtros dinámicos (por autor, por categoría, por año, etc.).
- **Evitar estado global compartido**: La información se pasa como parámetros de funciones y se devuelve como valores de retorno, reduciendo efectos secundarios.
- **Inmutabilidad lógica** (en lo posible): En lugar de modificar directamente slices globales, se pueden crear nuevos slices con los cambios aplicados y devolverlos.

## 2. Diseño del sistema

### 2.1 Diagrama General.

A continuación, se muestra el diagrama general de la función del programa.

Administrador ----> Gestión Libros ----> Base de Datos

L--> Gestión Usuarios

Usuario Consumidor ----> Consulta de Libros ----> Base de Datos

### 2.2 Flujo general de uso

- Administrador inicia sesión
- Admin gestiona libros y usuarios
- Usuario inicia sesión
- Usuario navega por el catálogo

## 3. Descripción detallada de los módulos

### 3.1 Módulo de usuarios

Incluye:

- Registro (solo admin)
- Editar usuario
- Dar permisos

- Eliminar usuario

### 3.2 Módulo de libros

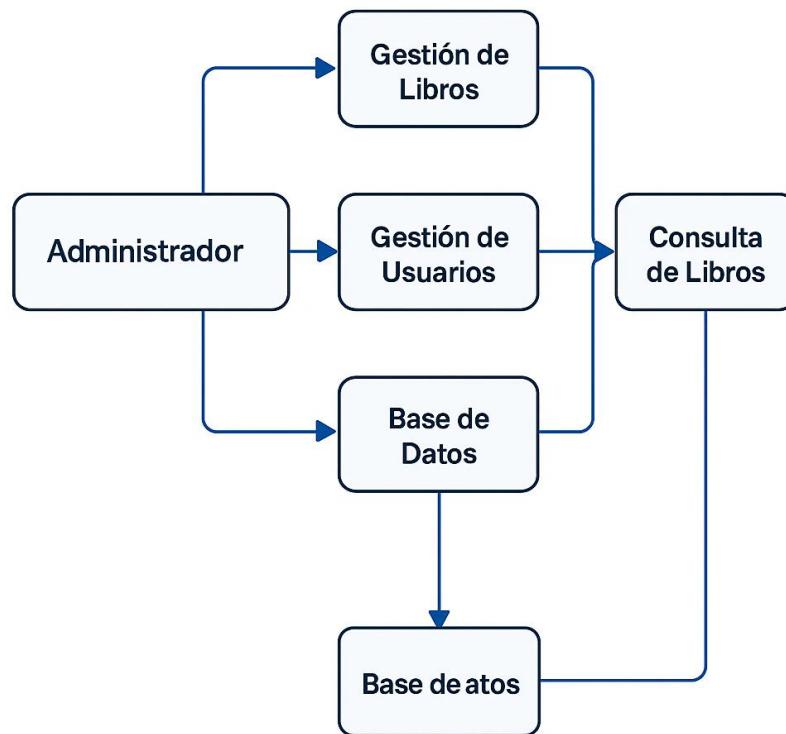
Incluye:

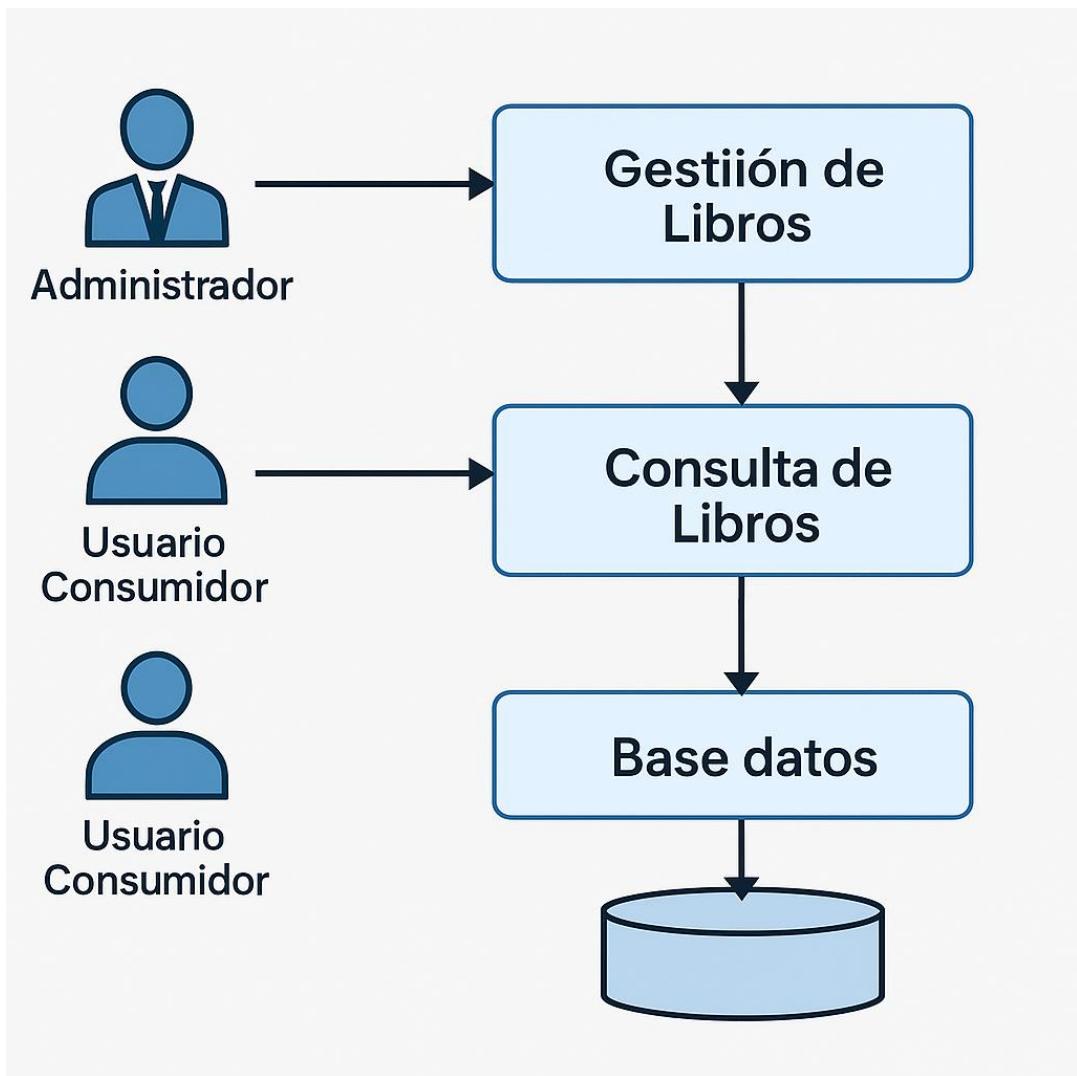
- Crear
- Editar
- Eliminar
- Listar

### 3.3 Módulo de búsqueda

- Buscar por autor
- Buscar por categoría
- Filtrar por año

### Diagramas





Link del video explicativo

[https://youtu.be/\\_gDVAwCB5no](https://youtu.be/_gDVAwCB5no)

Link del proyecto en github

<https://github.com/lenincube/sistema-libros>