1. Manual de Usuario – Aplicación de Escritorio (WinForms)

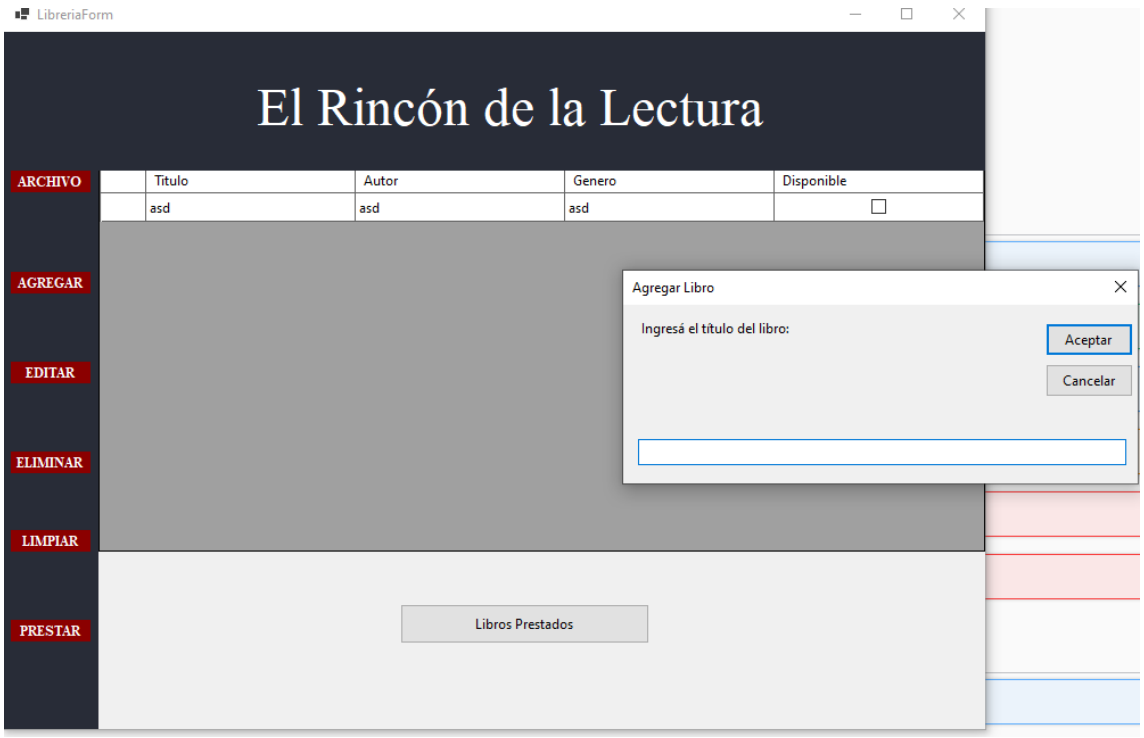
Este sistema de escritorio permite gestionar la información de una biblioteca. Desde la interfaz gráfica desarrollada en Windows Forms, el usuario puede realizar operaciones CRUD (crear, leer, actualizar y eliminar) sobre los registros de:

- Libros
- Autores
- Categorías
- Préstamos y Devoluciones

Interfaz principal:

La pantalla principal se titula **"El Rincón de la Lectura"**. Incluye:

- **Una tabla (DataGridView)** para visualizar libros.
- **Un panel lateral** con botones para agregar, editar, eliminar, limpiar y prestar libros.
- **Un botón inferior** para ver los libros prestados y registrar devoluciones.



Manual del Programador (Documento técnico)

Nombre del Proyecto:
LibreriaApi

Descripción General:

Este sistema está diseñado para gestionar una biblioteca. Permite administrar libros, autores, préstamos de libros y devoluciones. La aplicación está organizada bajo una arquitectura de **tres capas**, separando claramente responsabilidades:

- **API REST (ASP.NET Core)** para exponer endpoints que manejan la lógica del sistema.
- **Biblioteca de modelos compartida (`Libreria.Modelos`)** que define las entidades del dominio.
- **Aplicación de escritorio (WinForms)** que consume la API mediante `HttpClient`.

□ Arquitectura:

Se sigue una **arquitectura en capas** (API - Servicios - Modelos), aplicando principios **SOLID** para mantener un diseño limpio, mantenible y escalable. Se utiliza **Entity Framework Core** como ORM para la persistencia en base de datos.

Estructura del Proyecto:

`Libreria.Modelos`

Contiene las clases principales que representan las entidades del dominio:

- `Libro`: Título, autor, disponibilidad, etc.
- `Autor`: Nombre, nacionalidad.
- `Prestamo`: Relación entre usuario y libro prestado, con fecha de préstamo y devolución.
- `Usuario`: Persona que toma prestado un libro.
- `Categoria`: Agrupación de libros por temas.

`LibreriaApi` (API REST ASP.NET Core)

Carpetas principales:

- `Controllers/`:
  Exponen los endpoints HTTP para cada entidad (Libros, Usuarios, Préstamos, etc.).

- `Services/`:
  Contienen la lógica de negocio, separada de los controladores.
- `Interfaces/`:
  Interfaces que definen los contratos de los servicios, facilitando el desacoplamiento y testeo.
- `Data/DbContexto.cs`:
  Clase que representa el contexto de Entity Framework para la base de datos. Aquí se definen los `DbSet<>` de las entidades.
- `Migrations/`:
  Carpeta generada por EF Core con los scripts de migración de la base de datos.
- `Program.cs`:
  Configuración del servidor, servicios, CORS, Swagger y la inyección de dependencias.

`LibreriaDesktop` (WinForms)

Formularios y funcionalidades principales:

- `LibreriaForm.cs`:
  Formulario principal donde se centraliza la gestión de libros y préstamos.
- `models/`:
  Clases locales que reflejan las del backend para permitir el binding de datos en los formularios.
- La comunicación con la API se realiza mediante `HttpClient`.
- Se permite **prestar** y **devolver** libros desde la misma interfaz. Al devolver, se calcula si hay demora y se informa al usuario.

□ Tecnologías Utilizadas:

- **C# (.NET 8)**
- **Entity Framework Core**
- **Windows Forms (WinForms)**
- **REST API con Swagger**
- **Arquitectura en Capas con principios SOLID**
- **SQL Server**

3. Diagrama de Arquitectura (estructura)

```
css
CopiarEditar
```

```
LibreriaApi.sln
│
├── LibreriaApi/              ← API REST
│   ├── Controllers/          ← Controladores de entidades
│   ├── Services/             ← Lógica de negocio
│   ├── Interfaces/           ← Contratos de servicio
│   ├── Models/               ← Entidades (Libro, Prestamo)
│   ├── Data/DbContexto.cs    ← EF DbContext
│   └── Migrations/           ← Migraciones EF
│
└── LibreriaDesktop/          ← Aplicación WinForms
    ├── Forms/                ← Formularios: Principal, Préstamos
    ├── models/               ← Clases espejo del backend
    └── Program.cs            ← Entrada de app
```

4. Diagrama de Negocio (explicación para usuarios)

Este sistema permite:

- Registrar libros, autores, categorías y usuarios.
- Gestionar préstamos y devoluciones de libros.
- Consultar información general de la biblioteca desde una aplicación de escritorio.
- Controlar fechas de préstamos, devoluciones y calcular penalizaciones por retraso si corresponde.

Flujo del usuario:

1. El usuario abre la aplicación de escritorio (**LibreriaDesktop**).
2. Desde el menú principal, puede:
    - Ver la lista de libros disponibles y sus detalles.
    - Registrar nuevos libros, autores o categorías.
    - Registrar usuarios (lectores).
    - Realizar préstamos de libros a los usuarios.
    - Registrar devoluciones y calcular si hubo demoras.
3. Toda la información se almacena en una base de datos SQL y se accede en tiempo real a través de la **API REST**.