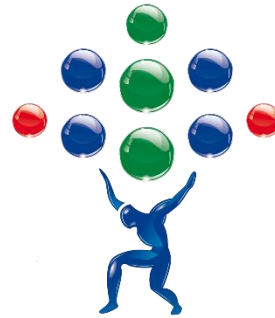


UNIVERSIDAD PRIVADA DOMINGO SAVIO

FACULTAD DE INGENIERIA



UNIVERSIDAD PRIVADA
DOMINGO SAVIO

TRABAJO:

PROYECTO FINAL

SISTEMA RESERVA DE PASAJES

ESTUDIANTES:

Jose Alejandro Benites Torrez

Alejandro Fernandez

DOCENTE:

ING. Yanet Colque Alarcon

TARIJA, Septiembre 2025

Sistema de Gestión de Biblioteca para Entornos Universitarios

Resumen

Este documento describe el diseño e implementación de un Sistema de Gestión de Biblioteca desarrollado con ASP.NET Core MVC, utilizando una arquitectura en capas. El sistema optimiza la administración de recursos bibliográficos, miembros, préstamos, reservas y eventos en entornos universitarios. Se detalla la estructura del sistema, incluyendo sus componentes principales (modelos, vistas, controladores) y funcionalidades como gestión de inventario, administración de usuarios y generación de reportes. Se incluye un ejemplo práctico de gestión de libros para ilustrar su aplicación en un contexto académico. El proyecto sigue principios de ingeniería de software, como la separación de responsabilidades y la inyección de dependencias, para garantizar escalabilidad y mantenibilidad.

Palabras clave: sistema de gestión de biblioteca, ASP.NET Core MVC, arquitectura en capas, entornos universitarios, gestión de recursos bibliográficos.

1. Introducción

La gestión eficiente de bibliotecas universitarias es crucial para apoyar las actividades académicas y de investigación en instituciones de educación superior. Davenport (1997) argumenta que los sistemas de información eficaces mejoran la gestión de recursos y la experiencia del usuario al reducir la fricción en los procesos operativos. Este documento presenta un Sistema de Gestión de Biblioteca desarrollado con ASP.NET Core MVC, estructurado en una arquitectura en capas para garantizar modularidad, escalabilidad y facilidad de mantenimiento.

El sistema gestiona recursos bibliográficos (libros, periódicos, revistas), miembros (estudiantes, docentes, personal), préstamos, reservas, eventos y reportes administrativos. Utiliza tecnologías modernas como Entity Framework Core para el acceso a datos, AutoMapper para el mapeo de objetos y Bootstrap para una interfaz de usuario responsive. Este informe describe la estructura del sistema, su desarrollo y los resultados obtenidos, con

un enfoque en su aplicación en entornos universitarios. Además, se presenta un ejemplo de gestión de libros con datos específicos (títulos, años de publicación, ISBN y descripciones) para ilustrar su funcionalidad.

2. Objetivos

El sistema tiene los siguientes objetivos principales:

1. Facilitar la gestión automatizada de recursos bibliográficos en bibliotecas universitarias.
2. Optimizar los procesos de préstamo, reserva y administración de miembros.
3. Proveer un dashboard administrativo con reportes y métricas clave para la toma de decisiones.
4. Garantizar una experiencia de usuario accesible y eficiente mediante una interfaz web responsive.

3. Arquitectura del Sistema

El sistema sigue una arquitectura en capas, un patrón ampliamente reconocido en ingeniería de software (Fowler, 2002). Las capas principales son:

- Library.Models: Contiene las entidades del dominio, mapeadas a la base de datos mediante Entity Framework Core.
- Library.Repositories: Implementa el acceso a datos utilizando los patrones Repository y Unit of Work.
- Library.Services: Encapsula la lógica de negocio, promoviendo la reutilización y la separación de responsabilidades.
- Library.ViewModels: Define modelos de vista para la presentación de datos en la interfaz de usuario.
- LibraryManagementSystem: La aplicación web principal, que incluye controladores y vistas basadas en ASP.NET Core MVC.

4. Desarrollo del Sistema

4.1. Modelos

Los modelos representan las entidades del dominio y son fundamentales para la gestión de datos. A continuación, se presentan dos ejemplos de código que ilustran la estructura de los modelos:

```
```csharp
// Book.cs

public class Book : LibraryItem
{
 public string ISBN { get; set; }
 public string Edition { get; set; }
}

// LibraryItem.cs (Clase base)

public abstract class LibraryItem
{
 [Key]
 public int Id { get; set; }
 [StringLength(12)]
 public string ItemCode { get; set; }
 [Required(ErrorMessage = "Enter a Title.")]
 [StringLength(100)]
 public string Title { get; set; }
 public int? PublishedYear { get; set; }
 public ItemType ItemType { get; set; }
}
```
```

Otros modelos incluyen Author, Member, Borrowing, LibraryEvent, Category, Genre, Language y Publisher, que permiten una gestión integral de los recursos y usuarios de la biblioteca.

4.2. Vistas

Las vistas están organizadas en áreas de ASP.NET Core MVC para diferenciar los roles de usuario:

- Admin: Gestiona libros, miembros, préstamos y reportes.
- Member: Permite a usuarios registrados buscar recursos, gestionar reservas y recibir notificaciones.
- Guest: Ofrece acceso público a información general y eventos.
- Identity: Maneja la autenticación y el registro de usuarios.

Las vistas utilizan Razor para generar HTML dinámico, con ViewModels que facilitan la transferencia de datos desde los controladores. Por ejemplo, la vista Home/Index.cshtml muestra eventos recientes y contenido destacado.

4.3. Controladores

Los controladores coordinan las solicitudes HTTP, interactuando con los servicios y repositorios. A continuación, se presenta un ejemplo de un controlador en el área administrativa:

```
```csharp
// BookController.cs (Admin Area)
[Area("Admin")]
public class BookController : Controller
{
 private readonly IBookService _bookService;
 public BookController(IBookService bookService)
 {
 _bookService = bookService;
 }

 public IActionResult Index(string searchTerm, int pageNumber = 1)
 {

```

```

 PagedResult<BookViewModel> result = _bookService.GetAll(pageNumber);
 return View(result);
 }
}
'''

```

Otros controladores incluyen AuthorController, BorrowingController, MemberController, DashboardController y ReportController, cada uno diseñado para manejar aspectos específicos del sistema.

## 5. Ejemplo de Gestión de Recursos: Libros

Para ilustrar la funcionalidad del sistema, se presenta una lista de libros gestionados, incluyendo títulos, años de publicación, códigos ISBN y descripciones breves (véase Tabla 1). Estos datos se integran en el sistema mediante la entidad Book, permitiendo operaciones CRUD (Crear, Leer, Actualizar, Eliminar) a través de la interfaz administrativa.

Tabla 1

Ejemplo de libros gestionados en el sistema

Título	Año	ISBN	Descripción
Cien años de soledad	1967	978-84-376-0494-7	Novela de Gabriel García Márquez que narra la saga de la familia Buendía...
1984	1949	978-0-452-28423-4	Distopía de George Orwell que describe un régimen totalitario...
Sapiens: De animales a dioses	2011	978-84-9992-622-3	Ensayo de Yuval Noah Harari que recorre la historia de la humanidad...
La casa de los espíritus	1982	978-84-01-35162-4	Novela de Isabel Allende que narra la historia de la familia Trueba...
El nombre del viento	2007	978-84-01-33720-8	Novela fantástica de Patrick Rothfuss que sigue las aventuras de Kvothe...

## 6. Resultados y Funcionalidades

El sistema implementa las siguientes funcionalidades clave:

1. Gestión de Inventario: Operaciones CRUD para libros, periódicos y revistas, con clasificación por categorías, géneros, idiomas y editores. Generación automática de códigos de ítem.
2. Administración de Miembros: Registro, gestión y seguimiento de estudiantes, docentes y personal, con historial de préstamos y reservas.
3. Préstamos y Reservas: Gestión de préstamos con cálculo de fechas de devolución y multas, y sistema de reservas para ítems no disponibles.
4. Eventos: Creación y gestión de eventos académicos, con registro de participantes y notificaciones automáticas.
5. Reportes y Dashboard: Panel administrativo con estadísticas y reportes sobre inventario, préstamos y membresías, utilizando gráficos generados con herramientas como Chart.js.
6. Notificaciones: Sistema de alertas por correo electrónico para recordatorios de devolución y eventos próximos.

El sistema utiliza inyección de dependencias, el patrón Unit of Work para transacciones y AutoMapper para el mapeo eficiente entre modelos y ViewModels. La interfaz de usuario, basada en Bootstrap, es responsive y accesible, cumpliendo con estándares de usabilidad para entornos académicos.

## 7. Conclusiones

El Sistema de Gestión de Biblioteca desarrollado es una solución robusta y escalable para la administración de bibliotecas universitarias. Su arquitectura en capas y el uso de tecnologías modernas garantizan un sistema mantenible y adaptable a las necesidades de una institución académica. La integración de un ejemplo práctico con datos de libros demuestra la aplicabilidad del sistema en la gestión de recursos bibliográficos. Futuras mejoras podrían incluir la integración con sistemas de autenticación universitaria (como LDAP) y la implementación de inteligencia artificial para recomendaciones personalizadas.

## Referencias

Davenport, T. H. (1997). Information ecology: Mastering the information and knowledge environment. Oxford University Press.

Fowler, M. (2002). Patterns of enterprise application architecture. Addison-Wesley.