

Mi Biblioteca

Explora los libros disponibles

[Inicio](#) [Libros](#) [Perfil](#) [Registrarse](#)

Información sobre la aplicación web

Esta plataforma ha sido desarrollada como un sistema integral de gestión de biblioteca que permite a los usuarios explorar libros, consultar información detallada, registrarse, iniciar sesión y navegar entre distintas secciones de forma intuitiva y segura.

El sistema combina una interfaz web amigable con un backend robusto, garantizando que tanto usuarios como administradores puedan interactuar con los datos de manera eficiente. La aplicación está diseñada siguiendo principios de arquitectura limpia, separando claramente las capas de dominio, aplicación e infraestructura.

Objetivo del proyecto

El propósito principal es ofrecer una experiencia clara y funcional para la consulta y gestión de libros, integrando una interfaz web dinámica con una API REST bien definida. De esta forma, se facilita tanto la interacción de los usuarios como la administración por parte del personal autorizado.

Además, se busca que el sistema sea fácilmente extensible: se pueden añadir nuevos módulos (por ejemplo, gestión de préstamos o reseñas de libros) sin alterar la estructura existente.

Características principales

- Visualización completa del catálogo de libros disponibles con imágenes, precios y descripciones.
- Búsqueda avanzada por título, autor, género u otros criterios.
- Acceso al detalle de cada libro con información ampliada.
- Registro e inicio de sesión de usuarios con validación de datos.
- Navegación clara mediante un menú superior y rutas bien definidas.
- API REST documentada mediante Swagger para facilitar integraciones externas.
- Gestión de ventas y compras con carrito de la compra integrado.
- Panel de administración para gestionar usuarios, libros y ventas.

Funcionamiento de la aplicación web

La aplicación está construida sobre **Spring Boot** en el backend y utiliza **Thymeleaf** como motor de plantillas para generar las vistas HTML. La estructura está organizada para separar claramente las vistas públicas, las de usuario cliente y las vistas exclusivas para administradores.

El flujo de datos sigue un patrón bien definido: los controladores reciben las peticiones HTTP, invocan los casos de uso de la capa de aplicación, convierten los modelos de

dominio a DTO mediante mappers y finalmente envían los datos al Model para que Thymeleaf los renderice en la vista.

Estructura de vistas

- **public/**: contiene las páginas accesibles sin autenticación (`login.html`, `register.html`, `allBook.html`, `details_book.html`).
- **client/**: vistas destinadas a usuarios con rol CLIENTE (`perfil.html`, `details_sales.html`, `edit_profile.html`, `edit_password.html`).
- **admin/**: vistas exclusivas para administradores, organizadas por módulos:
 - **book/**: gestión de libros (`list.html`, `form.html`).
 - **sale/**: administración de ventas (`list.html`, `new.html`, `detail.html`).
 - **users/**: gestión de usuarios (`users_list.html`, `formusers.html`).
 - `perfil.html`: perfil del administrador.
- **Raíz**: páginas generales como `index.html`, `contacto.html`, `informacionApi.html` e `informacionWeb.html`.

Controladores y rutas principales

En el paquete `infrastructure.adapter.in.web` se ubican los controladores que enlazan las rutas HTTP con las vistas:

- **HomeController**: gestiona las rutas principales (inicio, información, contacto).
- **BookController**: rutas públicas para ver libros y detalles.
- **AdminBookController**: módulo de administración de libros.
- **AdminUserController**: administración de usuarios.
- **AdminSaleController**: módulo de gestión de ventas.
- **ClientSaleController**: área de usuario cliente (perfil, compras, edición).
- **AuthController**: autenticación, registro y cambio de contraseña.

Cómo se conectan controladores y vistas

1. El usuario accede a una ruta (por ejemplo: `/web/admin/books`).
2. El controlador correspondiente procesa la petición y obtiene los datos desde los casos de uso.
3. Estos datos son convertidos a DTO mediante los mappers y enviados al Model.
4. Thymeleaf recibe los datos y renderiza la vista solicitada.
5. El usuario visualiza la información en la interfaz web y puede interactuar con ella.

Roles y seguridad

La seguridad está gestionada mediante **Spring Security**, aplicando restricciones según rol:

- Rutas `/web/admin/**`: solo accesibles para usuarios con rol **ADMIN**.
- Rutas `/web/client/**`: accesibles para roles **CLIENTE** y **ADMIN**.
- Rutas públicas: accesibles sin autenticación (`/web/books`, `/login`, `/register`, `/`).

Además, se aplican medidas de seguridad adicionales como encriptación de contraseñas, validación de formularios y protección contra ataques comunes (CSRF, XSS).

Resumen final

Esta aplicación integra Spring Boot, Thymeleaf y una API REST para ofrecer una plataforma completa de gestión bibliotecaria. Los usuarios pueden registrarse, ver su perfil y revisar sus compras; los administradores pueden gestionar libros, usuarios y ventas; y cualquier visitante puede explorar el catálogo sin necesidad de autenticación.

Gracias a su arquitectura modular y a la separación de responsabilidades, el sistema es mantenible, escalable y adaptable a nuevas necesidades. Esto garantiza que la biblioteca pueda evolucionar con el tiempo sin comprometer la estabilidad de la aplicación.

© 2025 Mi Biblioteca. Todos los derechos reservados.

[Información Web](#) | [Información API](#) | [GitHub](#)