
1. DESCRIPCIÓN Y OBJETIVOS GENERALES DEL PROYECTO

La aplicación Biblioteca es un sistema de gestión de libros, usuarios y ventas, desarrollado con Spring Boot, Thymeleaf y JPA/Hibernate. Se despliega en Docker y ofrece tanto vistas web como endpoints REST.

1.1 OBJETIVOS

- Crear una plataforma web funcional y estable.
- Gestionar libros, usuarios y pedidos desde vistas Thymeleaf.
- Implementar endpoints REST para clientes externos.
- Mantener arquitectura clara: controlador → servicio → repositorio → base de datos.
- Proveer autenticación con usuario administrador preconfigurado.
- Facilitar despliegue mediante Docker.

1.2 VERSIONES POSTERIORES

- Integración de JWT/OAuth2.
- Módulo de estadísticas y panel avanzado para administrador.
- Mejoras de usabilidad en la interfaz.
- Integración con servicios externos (notificaciones).
- API REST ampliada con paginación, filtros y Swagger.

1.3 REQUISITOS

- Java 17+
- Spring Boot
- Maven
- MySQL/H2
- Docker opcional
- Navegador moderno

2. JUSTIFICACIÓN DEL PROYECTO

Digitaliza la gestión de bibliotecas/librerías, permitiendo:

- Controlar inventario.
- Gestionar usuarios y roles.
- Administrar pedidos/ventas.
- Reducir errores manuales.
- Mejorar accesibilidad desde cualquier dispositivo.

3. PLANIFICACIÓN Y COSTE

Horas estimadas:

- Análisis: 20 h
- Diseño: 80 h
- Programación: 40 h
- Pruebas: 20 h
- Documentación: 2 h

TOTAL: 162 h

Recursos:

- PC de desarrollo
 - IDE (IntelliJ/Eclipse/VSCode)
 - Docker Desktop
 - Servidor MySQL/H2
 - GitHub
-

4. BASES DE DATOS

Entidades:

- Libro
- Usuario
- Pedido/Venta
- Rol

Relaciones:

- Usuario 1..n Pedidos
- Libro 1..n Pedidos

5. CÓDIGO FUENTE Y ORGANIGRAMAS

src/main/java

- controllers web (Thymeleaf)
- controllers REST
- services
- repositories
- models/entities

src/main/resources/templates → vistas Thymeleaf

src/main/resources/static/css → estilos

src/main/resources/static/img → imágenes

Flujo:

1. Usuario solicita vista.
2. Controlador carga datos.
3. Servicio aplica lógica.
4. Repositorio accede a BD.
5. Thymeleaf renderiza HTML.

6. PRUEBAS REALIZADAS

Usuario admin:

- email: admin33@admin.com
- contraseña: 123
- rol: ADMIN

Pruebas:

- CRUD libros
- Registro/login
- Validaciones de algunos formularios

- Endpoints REST
 - Navegación vistas admin/cliente
-

7. MANTENIMIENTO

- Revisión de logs semanales.
 - Actualización dependencias Maven.
 - Control crecimiento BD.
 - Refuerzo seguridad y validaciones.
-

8. MANUALES

a. MANUAL DE USUARIO

- i. Introducción → Navegar catálogo, registrarse, iniciar sesión, pedidos.
- ii. Registro/login → Pantallas /login y /register.
- iii. Administración → Acceso /admin, gestión libros/usuarios/pedidos.
- iv. Compra → Desde detalle de libro.
- v. Finalizar pedido → Confirmación y registro en BD.

b. MANUAL DE LA APLICACIÓN

- i. Introducción → Guía técnica backend.
- ii. Requisitos → Java, Maven, BD, Docker.
- iii. Instalación → mvn clean install, mvn spring-boot:run.
- iv. Gestión de datos → Repositorios JPA CRUD.
- v. Interfaz → Plantillas Thymeleaf en /templates, CSS en /static/css.
- vi. Resolución problemas → Puerto ocupado, error BD, Docker.

c. GUÍA DE ESTILO

- i. Importancia comunicación visual → Vistas claras.
- ii. Colores/tipografías → Definidos en CSS.
- iii. Presentación → Secciones, títulos, tarjetas.
- iv. Aplicación → Consistencia en pantallas.
- v. Adaptabilidad → Responsive, móvil.
- vi. Bootstrap → Opcional, CSS propio.
- vii. Creatividad → Diseños coherentes.
- viii. Redes sociales → Futuro.
- ix. Satisfacción/test calidad → Pruebas funcionales.
- x. Imágenes → /static/img.
- xi. Vídeos/audios → No aplicable.
- xii. Animaciones/GIFs → No incluidas, integrables.

9. CASO PRÁCTICO

Usuario inicia sesión, navega catálogo, añade libro al pedido, confirma compra, revisa historial. Admin visualiza pedidos en panel.

10. BIBLIOGRAFÍA

-
- Documentación Spring Boot
 - Documentación Maven
 - JPA/Hibernate ORM
 - Docker Docs
 - README.md del proyecto