

# Informe del Sistema de Gestión de Bibliotecas

Tú Nombre

November 30, 2023

## 1 Introducción

El Sistema de Gestión de Bibliotecas (LMS) desarrollado tiene como objetivo principal gestionar eficientemente las operaciones de una biblioteca pública grande, como préstamos, devoluciones, renovaciones, adición y eliminación de libros. Se utiliza Hibernate y JPA para interactuar con una base de datos SQL y OpenAPI/Swagger UI para proporcionar una API REST.

## 2 Modelo de Datos

El modelo de datos se ha diseñado teniendo en cuenta las entidades clave de una biblioteca: Libro, Lector y Préstamo. Además, se han incluido la entidad Bibliotecario para mejorar la organización y acciones para el manejo de la biblioteca. Las relaciones entre estas entidades se han establecido de manera coherente.

## 3 BBDD (PostgreSQL)

PostgreSQL ha sido seleccionada como base de datos de este proyecto, debido a que es una base de datos relacional de código abierto, que destaca por sus características avanzadas, extensibilidad, soporte para tipos de datos personalizados, lenguajes de programación PL/pgSQL, integridad referencial, manejo de concurrencia, y soporte para JSON. Ofrece optimización de consultas, seguridad robusta, escalabilidad, rendimiento mejorado, una comunidad activa y una licencia de código abierto.

## 4 Hibernate y JPA

Las entidades están anotadas con las anotaciones de JPA, permitiendo así que Hibernate mapee estas clases a las tablas correspondientes en la base de datos. Se han establecido tanto relaciones bidireccionales como unidireccionales entre las entidades para facilitar la navegación y consulta de datos.

## 5 OpenAPI/Swagger UI

La utilización de OpenAPI/Swagger UI como herramientas para el desarrollo de la API del proyecto, es por las siguientes razones: interfaz interactiva para probar APIs, validación de especificaciones, generación automática de código cliente y servidor, consistencia en el diseño de la API, facilitación de la colaboración entre equipos, soporte para versionamiento y evolución, y acceso a un amplio ecosistema de herramientas y servicios compatibles. Estas herramientas simplifican el desarrollo, mejoran la comprensión de la API y contribuyen a la coherencia en el desarrollo de software.

## 6 Docker Compose

La herramienta Docker Compose juega un papel crucial al orquestar servicios y componentes en contenedores. Así como JPA y Hibernate simplifican el mapeo objeto-relacional en Java, Docker Compose utiliza un archivo YAML para definir la configuración de contenedores y servicios, estableciendo relaciones y dependencias. Esta flexibilidad facilita la creación de entornos de desarrollo consistentes y reproducibles, contribuyendo a la eficiencia y mantenimiento de aplicaciones en entornos contemporáneos.

## 7 Frontend

En el ámbito del desarrollo frontend, la combinación de herramientas como Thymeleaf, Bootstrap y Webpack ofrece una potente sinergia para la creación de interfaces de usuario modernas y dinámicas. Thymeleaf, con su capacidad de plantillas en el lado del servidor, se fusiona armoniosamente con Bootstrap, un framework de diseño que simplifica la creación de interfaces responsivas. Webpack, por su parte, gestiona eficientemente las dependencias y optimiza los recursos, asegurando un rendimiento óptimo. En conjunto, estas herramientas establecen relaciones estructuradas entre las capas frontend y backend, permitiendo la construcción de aplicaciones web modernas con una experiencia de usuario robusta y agradable.

## 8 Conclusión

La implementación de este Sistema de Gestión de Bibliotecas demuestra un enfoque integral y bien estructurado, aprovechando tecnologías y herramientas modernas para lograr eficiencia, coherencia y una experiencia de usuario mejorada.