

1.1.1. LMS

El LMS es el núcleo del sistema. Es en este servicio donde se encuentra la lógica del sistema. Los controladores están implementados en Java utilizando el framework Spring Boot. Al utilizar el framework Spring Boot este permite desplegar la instancia al servicio LMS en un servidor externo, o en un Tomcat embebido. El LMS está implementado para servir diferentes tipos de peticiones, que están agrupadas en dos grupos diferenciados.

De forma que las URIs validas que acepta el LMS son de la siguiente forma:

- `/service/{query}`
- `/resource/{id_resource}[?{query}]`

El espacio URI incluye selectores parámetros de consulta. La instancia del LMS procesa las URIs para luego reenviarlas al módulo correspondiente para que sirva las respuestas, es decir, en caso que la petición sea para el WMS, es en el LMS donde se procesa la URI para reenviar la petición al WMS con el formato adecuado para que este pueda procesarla, y de igual manera con las peticiones que tendrán que ser enviadas al módulo de SPARQL.

1.1.2. API del LMS

La API del LMS está compuesta por las siguientes operaciones: lectura de recursos y edición/actualización de recursos. También haría las funciones de proxy inverso en la parte referente al WMS, ya que se encarga del redireccionamiento de las peticiones.

Las librerías utilizadas para el desarrollo de esta API son:

- **Spring Boot** es parte de Spring IO Platform, un framework que facilita el desarrollo de proyectos y su posterior testeo. Spring Boot es un componente importante dentro del LMS porque, se encarga de la configuración de los componentes, principalmente a través de la inversión de control, utiliza el modelo vista-controlador para la implementación de servicio web, utiliza el Tomcat embebido, y la herramienta de pruebas durante la implementación.
- **Apache Jena** es un framework de código abierto que permite trabajar en el ámbito de web semántica desde Java. Este framework ofrece operaciones para

trabajar con información en formato RDF mediante el uso de grafos RDF, dichos grafos se representan a través de modelos, que pueden generarse de diversas fuentes o incluso fuentes combinadas (ficheros, bases de datos, etc.) y soportan las consultas, mediante el lenguaje SPARQL.

Se puede encontrar la documentación completa del proyecto en <https://github.com/IAAA-Lab/LinkedMapService/blob/master/Memoria.pdf>