# **Tecnologías y protocolos adyacentes a Apache Marmotta.**

**Objetivo:** Comprender cómo funciona Apache Marmotta mediante el análisis de su código fuente y sus protocolos de funcionamiento.

**Resultados:** Diagramas que describan los elementos más generales que permiten que Apache Marmotta funcione.

Para el desarrollo de esta actividad, se decidió por apoyarse en el diagrama ofrecido por los desarrolladores de Apache Marmotta en su página oficial[[1]](#footnote-1).

Los programadores de Apache desarrollaron Marmotta inspirándose en la arquitectura orientada a servicios, *SOA* por sus siglas en inglés, basando su servicio en el *framework* CDI/Weld tales como los componentes núcleo de Java EE 6.0

Las características de SOA son las siguientes

* Objetivos de negocio ligados a la infraestructura TI
* Orientada a la arquitectura de sistemas. Esto busca que la lógica de procesos de negocio no intervenga con la lógica del software de un sistema.
* Separación de objetivos: Dividir objetivos primarios en diferentes características con funcionalidades estrechas tan pequeñas como sean posible.
* Modularidad: La aplicación estará dividida en piezas distinguibles las cuales, desempeñaran una función en específico en el sistema.
* Bajo acoplamiento: Los atributos de los componentes de un sistema no tienen y/o no hacen uso del conocimiento de otros componentes independientes.
* Encapsulación: El acceso a datos con sus respectivas instrucciones de manipulación, las cuales estarán dentro de un paquete, es posible mediante una interfaz independiente.
* Interfaces: Implementación de un pequeño conjunto de interfaces que son mantenidas de manera separada.
* Mensajes: Uso de mensaje que contengan información a ser intercambiadas a través de las interfaces mediante una estructura y vocabulario delimitado por un esquema.
* Reutilización: Es la acción de reutilizar un componentes múltiples veces.
* Composabilidad: Capacidad de seleccionar componentes y ensamblarlos de diversas maneras que puedan cumplir el objetivo de la aplicación.

En la figura 1 se muestra la arquitectura del software y una breve descripción de la plataforma. La arquitectura de Marmotta está constituida por 5 capas

* Capa de interfaz de usuario: Se implementó HTML5, JavaScript y llamadas tipo REST para acceder al servidor. Puesto que Marmotta es un servidor de aplicaciones, la interfaz de usuario solo es para interfaces de administración y desarrollo, por lo que no está dirigido para los usuarios finales. Las tecnologías usadas en esta capa son:
  + HTML5.
  + Javascript.
* Capa de servicio Web: La capa ofrece servicios web tipo REST para acceder a las funcionalidades del servidor. Dichos servicios Web usan y generan archivos JSON y/o datos tipo RDF. En Marmotta se usan JAX-RS y RESTeasy para implementar los servicios web. Así las tecnologías usadas en esta capa son:
  + REST
  + JSON
  + RDF
* Capa de servicios: Se ofrecen servicios CDI en el entorno de Java los cuales pueden ser invocados desde Java. Con base a la documentación, no se necesita hacer uso de estos servicios a menos que se desarrollen aplicaciones Java basándose en Marmotta. Por tanto, la tecnología usada es:
  + CDI / WELD: Es un *framework* implementado en Java desde la versión 6 el cual resuelve dependencias entre servicios que están dentro del estándar JEE. Este mecanismo permite administrar el ciclo de vida de componentes e inyectar[[2]](#footnote-2) componentes (servicios) en objetos de clientes de una manera segura[[3]](#footnote-3).
* Capa de modelo: Ésta capa ofrece persistencia y acceso a datos. Estas funciones son proveídas por la librerías comunes Java de Marmotta (ldcache, kiwi, ldclient, Repositorio Sesame entre otras plasmadas en la figura 1 en bloques color rojo).
  + Ldcache: Módulo que permite agregar diferenes configuraciones *cache* para las fuentes de datos registrados en *Ldclient*.
  + Kiwi *triple store*: Es el software de Apache Marmotta que permite llevar a cabo transacciones de alto rendimiento al *framework* de consultas, almacenamiento y análisis de datos RDF *OpenRDF Sesame* mediante la construcción de una base de datos relacional. Cabe decir que KiWi *triple store* es el *backend* de Apache Marmotta.
  + Ldclient: Es un módulo que ofrece la infraestructura para recibir información de fuentes remotas mediante diferentes protocolos, principalmente HTTP, de proveedores de datos tales como YouTube o Facebook, mapeando los datos en estructuras RDF apropiadas para ser un recurso de datos de tipo *Linked data.*
  + SPARQL Sesame: Es un *framework* *open source* que permite consultar, almacenar y analizar datos tipo RDF. Actualmente está adoptado por *Eclipse* recibiendo el nombre *Eclipse RDF4J[[4]](#footnote-4).*
* Capa de persistencia: Son plataformas *open source* ajenas, pero con la posibilidad de ser usadas por Marmotta para funcionar, tales como
  + PostgreSQL[[5]](#footnote-5): Es un sistema de bases de datos objeto-relacional que usa y extiende la funcionalidades del lenguaje SQL para dar soporte a cargas de datos considerables.
  + H2[[6]](#footnote-6): Es una base de datos SQL para Java. Las características de esta base de datos son
    - Es *open source*
    - Soporta el estándar SQL
    - Contiene mecanismos de seguridad
    - Puede ser usada mediante el controlador de *PostgreSQL.*
  + MySQL[[7]](#footnote-7): Es una base de datos relacional *open* *source* rápida, confiable, escalable y fácil de usar. Puede usarse en clientes, servidores y sistemas embebidos.

Cabe decir que las capas descritas anteriormente se comunican mediante diversos medios. En la capa de servicio, inyecciones de servicio tipo CDI y notificaciones de eventos son usados para enviar mensajes mientras que en la capa de persistencia se usan transacciones tipo *KiWi* enviándose notificaciones de actualizaciones de *triple store*.

Por otra parte, en la figura 1 también se observan los elementos que conforman a Marmotta

* *Core[[8]](#footnote-8):* Módulo que proporciona todas las funcionalidades para que la plataforma funcione. Los demás módulos dependen del *Core*.
* *LDCache[[9]](#footnote-9):* Módulo que permite agregar diferentes configuraciones *cache* para las fuentes de datos registrados dentro del módulo *Ldclient*.
* *LDPath[[10]](#footnote-10):* Lenguaje de consulta basado en rutas diseñado para consultar y recibir datos de la nube del *Linked Data* al seguir enlaces *RDF* entre los servidores y las fuentes de información.
* *Reasoner[[11]](#footnote-11):* Módulo capaz de llevar a cabo inferencias mediante secuencias lógicas basándose en el lenguaje ontológico usado en Marmotta.
* *SPARQL 1.1[[12]](#footnote-12):* Módulo de Marmotta que da soporte a *SPARQL 1.1*. La documentación de Marmotta menciona que de las 11 recomendaciones que *SPARQL 1.1* propone, solo se tienen implementadas 9. Esto implica que 2 están pendientes por cumplir, entre una de ellas, la capacidad de llevar a cabo consultas federadas.
* *Versioning[[13]](#footnote-13):* Servicio que Marmotta ofrece para mantener actualizada actualizaciones de tripletas mediante grafos configurables. Esto lo lleva a cabo mediante estampas de tiempo y *snapshots* del grafo.

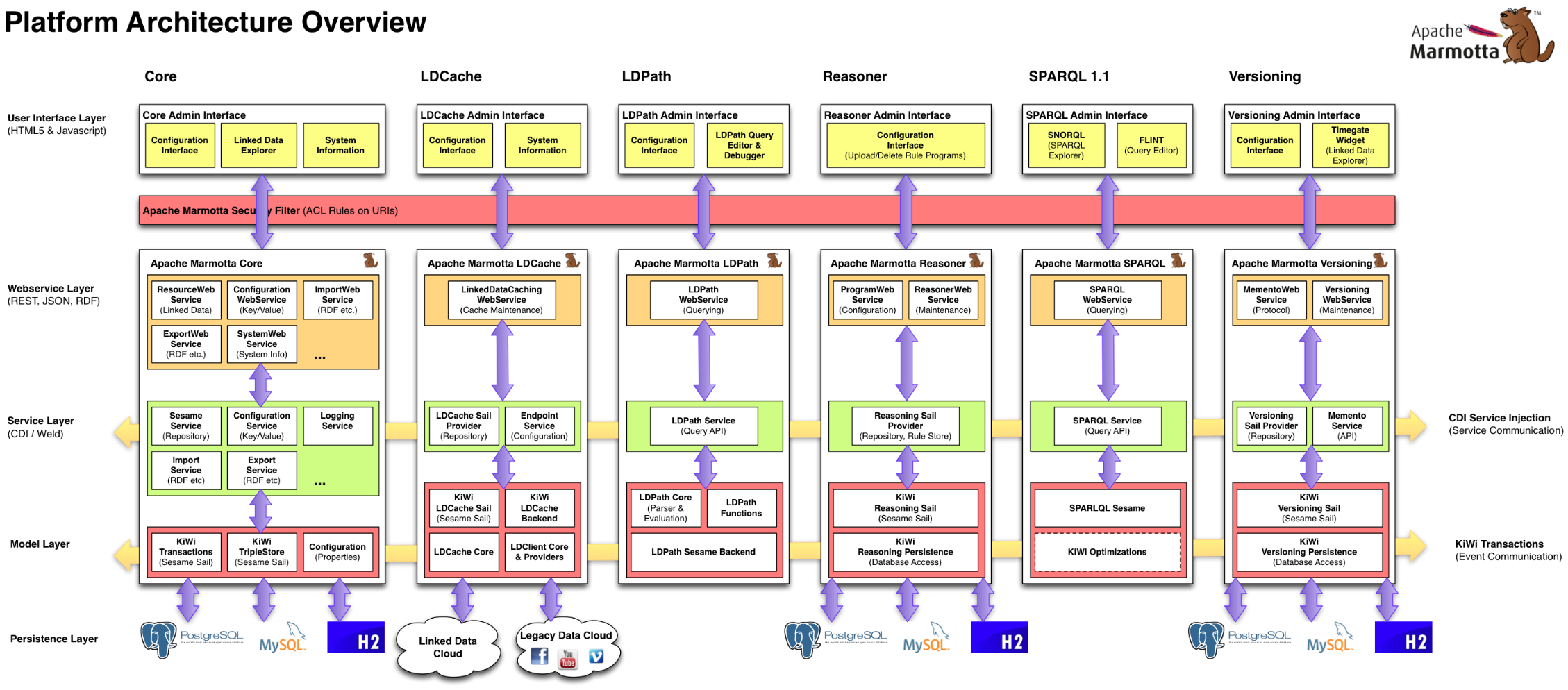


Fig. 1 Arquitectura de Apache Marmotta

1. <https://marmotta.apache.org/platform/index.html> [↑](#footnote-ref-1)
2. Se le denomina inyección, en ingeniería de software, a la técnica por la cual un objeto, o método estático, suministra las dependencias de otro objeto. Las dependencias son objetos que pueden ser utilizados, por lo que son también considerados como un servicio. [↑](#footnote-ref-2)
3. <https://weld.cdi-spec.org/> [↑](#footnote-ref-3)
4. <https://rdf4j.eclipse.org/documentation/migration/> [↑](#footnote-ref-4)
5. <https://www.postgresql.org/about/> [↑](#footnote-ref-5)
6. <https://www.h2database.com/html/main.html> [↑](#footnote-ref-6)
7. <https://dev.mysql.com/doc/refman/8.0/en/what-is-mysql.html> [↑](#footnote-ref-7)
8. <https://marmotta.apache.org/platform/core-module.html> [↑](#footnote-ref-8)
9. <https://marmotta.apache.org/platform/ldcache-module.html> [↑](#footnote-ref-9)
10. <https://marmotta.apache.org/platform/ldpath-module.html> [↑](#footnote-ref-10)
11. <https://marmotta.apache.org/platform/reasoner-module.html> [↑](#footnote-ref-11)
12. <https://marmotta.apache.org/platform/sparql-module.html> [↑](#footnote-ref-12)
13. <https://marmotta.apache.org/platform/versioning-module.html> [↑](#footnote-ref-13)