

Instituto de Computación
Facultad de Ingeniería
Universidad de la República

LOAD

Localizador de Objetos de Aprendizaje Distribuidos



Proyecto de Grado

Tutor: Dra. Regina Motz – Diego Vallespir

Autor: Rodrigo Saavedra

Resumen

El objetivo de esta tesis es proponer una arquitectura de software para la construcción de un sistema que permita la localización de objetos de aprendizaje de forma universal para poder integrarlos en un sistema de *e-learning*.

Los sistemas de aprendizaje utilizan objetos de aprendizaje como base del contenido de sus cursos. Estos objetos residen en repositorios. El objetivo de un repositorio es facilitar la reutilización de dichos recursos educativos, facilitando el acceso a los mismos.

Para que un objeto de aprendizaje sea reutilizado debe ser desarrollado de forma que se ajuste, al menos, a algún estándar de etiquetado de metadatos asociados a los contenidos que contenga.

En resumen, estamos ante una situación en que la arquitectura constituye un serio freno a la evolución en términos de desarrollo y accesibilidad. Se necesita una arquitectura realmente distribuida y fácilmente extensible, de forma que el sistema pueda evolucionar ante los continuos cambios en el área. Esta arquitectura se sustentará en los metadatos asociados a los objetos de aprendizaje que los convierten en piezas fundamentales del sistema, y en protocolos que harán posible cumplir con los objetivos buscados. En este punto, se plantea la construcción de una federación de repositorios de objetos de aprendizaje, hablando de federación como la integración e interoperabilidad de los repositorios para la localización de objetos de aprendizaje de forma universal.

La solución que proponemos es la definición de un marco funcional y arquitectónico para la adaptación de un sistema implementado en base a servicios Web (que proporciona un mecanismo flexible de integración de distintas aplicaciones y de descubrimiento de recursos en Internet), que asegure la interoperabilidad de distintos repositorios de objetos de aprendizaje y que favorezca la reutilización de los mismos.

Se propone una arquitectura en capas que ha de satisfacer una serie de requisitos que deberán observarse como condiciones fundamentales a considerar en cualquier sistema que se base en dicha arquitectura. También se definen los componentes necesarios de la arquitectura para asegurar la funcionalidad requerida, así como el flujo de información y las relaciones entre ellos.

Para validar la arquitectura propuesta se describirá un prototipo real creado a partir de los principios arquitecturales propuestos y desarrollado en su totalidad con software libre. El prototipo implementado trabaja con los metadatos representados en el esquema LOM, pero es fácilmente extensible al esquema de metadatos Dublin Core. Una de las principales funcionalidades, es el de la búsqueda federada sobre los repositorios de la federación, permitiendo especificar valores de búsqueda para cada uno de los metadatos de LOM.

Palabras clave:

E-learning, Objeto de aprendizaje, Repositorio, Federación, Estándar, Metadato, LOM, SCORM, SQI.

Índice de Contenidos

1.	Introducción.....	5
1.1.	Análisis y objetivos del Proyecto.....	8
1.2.	Estructura del documento.....	9
2.	Marco Conceptual.....	11
2.1.	Repositorios y objetos de aprendizaje.....	11
2.2.	Estándares de Metadatos en <i>e-learning</i>	12
2.2.1.	Tipos de estándares en <i>e-learning</i>	13
2.2.2.	Organizaciones de estandarización.....	14
2.2.3.	Metadatos.....	15
2.2.3.1.	LOM.....	15
2.2.3.2.	Dublin Core.....	17
2.2.3.3.	SCORM.....	18
2.3.	SQI (Simple Query Interface).....	20
2.3.1.	Lenguaje de consulta y formato de resultados.....	21
2.3.2.	Síncrono y Asíncrono.....	21
2.3.3.	Gestión de sesión.....	21
2.3.4.	<i>Stateless/Stateful</i>	22
2.3.5.	Principio de separación Comando-Consulta.....	22
2.3.6.	Métodos de SQI.....	22
2.4.	Servicios Web y Arquitectura SOA.....	23
2.4.1.	Descripción general de los Servicios Web.....	24
2.4.2.	Estándares y tecnologías.....	26
2.4.3.	Arquitectura Orientada a Servicios (SOA).....	27
2.4.4.	Servicios Web y Arquitectura Orientada a Servicios.....	28
3.	Relevamiento de Federaciones.....	30
3.1.	Estándar IMS DRI.....	30
3.2.	ARIADNE.....	33
3.3.	GLOBE.....	36
3.4.	AGREGA.....	37
3.5.	CORDRA.....	39
3.6.	Conclusiones.....	41
4.	Arquitectura Propuesta LOAD.....	43
4.1.	Introducción.....	43
4.2.	Funcionalidad soportada por la Arquitectura.....	43
4.2.1.	Especificación de requisitos.....	43
4.2.2.	Actores.....	44
4.2.3.	Casos de Uso.....	44
4.2.4.	Modelo de Dominio.....	48
4.3.	Organización en capas de la Arquitectura propuesta.....	49
4.3.1.	Capa de Datos y Sistemas existentes.....	50
4.3.2.	Capa de Comunicación.....	50
4.3.3.	Capa Lógica.....	51
4.3.4.	Capa de Acceso y Presentación.....	51
5.	Implementación del Prototipo.....	53
5.1.	Tecnologías.....	53
6.	Conclusiones y Trabajos Futuros.....	54
6.1.	Conclusiones.....	54
6.2.	Trabajos Futuros.....	54
7.	Bibliografía y referencias.....	57

Índice de Figuras

Figura 1 – Componentes en un entorno e-learning [7]	6
Figura 2– Componentes de los Objetos de Aprendizaje	11
Figura 3 – Diferentes estructuras de repositorios	12
Figura 4 – Áreas afectadas por los estándares en e-learning [8].....	13
Figura 5 – Estructura general de LOM.....	15
Figura 6 – Organización de las especificaciones SCORM [30]	20
Figura 7 – Modelo de comunicación SQI [13]	21
Figura 8 - Funcionamiento de los servicios Web [25]	24
Figura 9 – Esquema de funcionamiento de la IMS DRI [7].....	31
Figura 10 – Interfaz de consulta de ARIADNE	33
Figura 11 – Distribución de ARIADNE.....	34
Figura 12 – Componentes de la Arquitectura de ARIADNE [17].....	34
Figura 13 – Búsqueda Federada de Ariadne [26]	35
Figura 14 – API SQI a través de servicios Web [26]	35
Figura 15 – Interfaz de consulta GLOBE.....	36
Figura 16 – Arquitectura de AGREGA [19].....	39
Figura 17 – Modelo de Dominio del problema.....	48
Figura 18 – Estructura en Capas de la Arquitectura.....	49
Figura 19 – Mapeo Dublin Core a LOM.....	56

Índice de Tablas

Tabla 1 - Componentes de LOM	17
Tabla 2 – Componentes de Dublin Core	18
Tabla 3 – Métodos de la API SQI.....	23
Tabla 4 – Conjunto de funciones de la IMS DRI	32
Tabla 5 – cuadro comparativo de federaciones.....	41
Tabla 6 – Requisitos funcionales	44
Tabla 7 – Plantilla de casos de uso	44

1. Introducción

Actualmente se vive en una era en donde la información y el conocimiento implican el desarrollo de una sociedad globalizada, donde se atraviesan fronteras y límites geográficos, a esto se lo conoce como “sociedad de la información” [3]. La información es el nuevo y preciado recurso al cual se debe proporcionar la capacidad de acceso de forma universal. Esta cantidad de información que caracteriza la sociedad moderna impone más que nunca la necesidad de aprender; sin embargo, un volumen importante de información a aprender y la velocidad requerida para hacerlo pueden llegar a ser un importante obstáculo. Es por esto que muchos autores denominan a esta sociedad como: “la sociedad del aprendizaje” [3].

Hasta el momento los libros han definido la forma de aprendizaje en los centros educativos de la sociedad moderna. Con la llegada de los computadores la información pasó a tener un formato digital y cuando se empezaron a construir las primeras redes de computadores esta información empezó a ser compartida por distintas personas que utilizaban la misma red. Con el paso del tiempo la innovación tecnológica ha conseguido que la difusión mundial del conocimiento y de la información sea algo natural. Este medio de comunicación es Internet.

Las tecnologías de la información están cambiando el acceso al conocimiento, los procesos de aprendizaje y los diferentes procedimientos establecidos dentro del campo de la educación. Por lo tanto, para que la educación sea más eficiente debe adaptarse a los requerimientos actuales.

Toda esta transformación, conduce a que hoy en día se hable de *e-learning* (aprendizaje electrónico, del inglés “electronic learning”) como sinónimo de educación a través de las tecnologías de la información. Los *e-learning* son sistemas de información complejos en los que intervienen usuarios con distintas habilidades y objetivos, sistemas de diversas aplicaciones con tecnologías heterogéneas y contenidos digitales de formas y formatos diversos. Llegar a una operación e integración en la que todos los componentes involucrados (personas y sistemas) realicen sus tareas, interactúen y obtengan los resultados deseados es una tarea muy compleja que requiere de esfuerzos importantes para lograr la comunicación y la transmisión de información.

Hablar hoy en día de *e-learning* implica hablar de un entorno donde los contenidos, las actividades y recursos del aprendizaje se encuentran estructurados en objetos de aprendizaje, que se almacenan en Repositorios de objetos de aprendizaje y se administran en Sistemas de Administración del aprendizaje, apoyados en Sistemas de Gestión del Contenido. Este entorno implica también hablar de distintos usuarios: tutores, estudiantes, autores o creadores de contenidos, pedagogos o diseñadores instruccionales y expertos en TICs. Los componentes involucrados en un entorno de e-learning se describen como muestra la Figura 1:

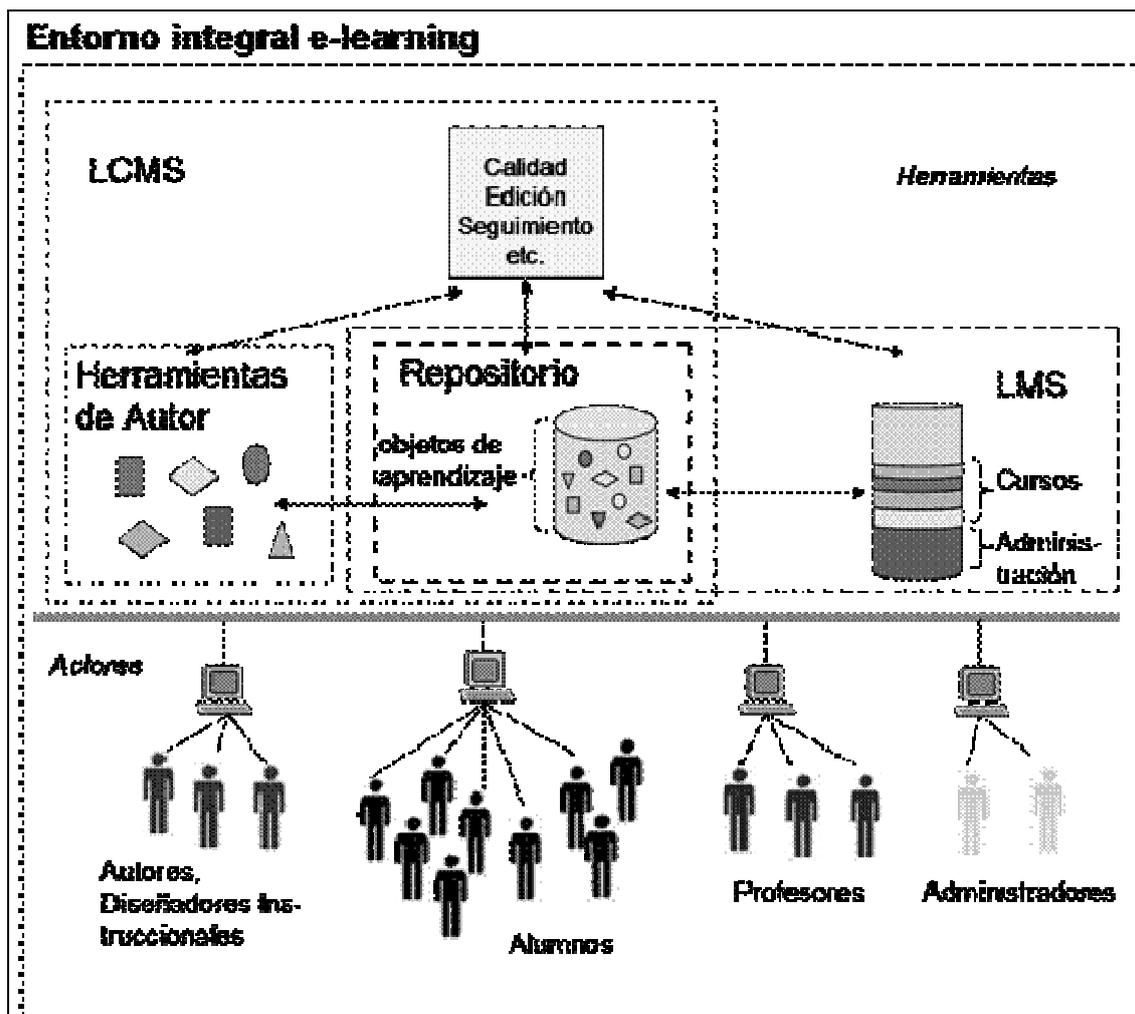


Figura 1 – Componentes en un entorno e-learning [7]

Entre las herramientas más utilizadas para los ambientes o sistemas *e-learning* están los Sistemas de Administración de Aprendizaje o **LMS** (Sistemas de Gestión de Aprendizaje, del inglés "*Learning Management Systems*"). Constituyen una categoría de software que automatiza la administración de acciones de formación: gestión de usuarios, gestión y control de cursos, gestión de los servicios de comunicación, etc. Estos sistemas gestionan los contenidos almacenados generalmente en repositorios. Presentan algunas carencias (poca formalización y estructuración del conocimiento, poca adaptabilidad a los estándares, contenidos demasiado expositivos, etc.) que los configuran como herramientas con limitaciones en su interacción con los alumnos.

Los **LCMS** (Sistemas de Gestión de Contenidos, del inglés "*Learning Content Management Systems*") se definen como un sistema basado en Web que es utilizado para crear, aprobar, publicar, administrar y almacenar recursos educativos (por ej. objetos de aprendizaje) y cursos en línea.

En este contexto surge el concepto de objeto de aprendizaje. Formalmente no hay una única definición del concepto de objeto de aprendizaje y las definiciones son muy amplias. El Comité de Estandarización de Tecnología Educativa IEEE (IEEE LTSC, del inglés "*Learning Technology Standards Committee*"), define a los objetos de aprendizaje (OA) como "una entidad, digital o no digital, que puede ser utilizada, reutilizada y referenciada durante el aprendizaje apoyado con tecnología".

Los objetos de aprendizaje brindan a las docentes posibilidades de interoperabilidad del material de forma independiente a la plataforma de administración del mismo y también posibilidades de reutilización en distintos contextos. Sin embargo, la capacidad de reutilización depende de la labor que desarrolle el creador de contenidos didácticos para generar “unidades mínimas en las que pueda organizar el material de formación para facilitar la gestión del conocimiento: creación, indexación, almacenamiento, distribución, uso, reutilización, evaluación y mejora de la formación” [3].

Un OA no puede ser creado como otro recurso más de información aislada, en su concepción debe pensarse que sean recursos con atributos específicos para su interacción en un entorno e-learning, fácil de localizar, utilizar, almacenar y compartir. Para ello, estos recursos deben ser:

- *Reutilizables*: El recurso debe ser modular para servir como base o componente de otro recurso. También debe tener una tecnología, una estructura y los componentes necesarios para ser incluido en diversas aplicaciones.
- *Accesibles*: Pueden ser indexados para una localización y recuperación más eficiente, utilizando esquemas estándares de metadatos.
- *Interoperables*: Pueden operar entre diferentes plataformas de hardware y software.
- *Portables*: Pueden moverse y albergarse en diferentes plataformas de manera transparente, sin cambio alguno en estructura o contenido.
- *Durables*: Deben permanecer intactos a las actualizaciones (*upgrades*) de software y hardware.

Los metadatos por su parte son un conjunto de atributos o elementos necesarios para describir un objeto de aprendizaje. A través de los metadatos se tiene un primer acercamiento con el objeto, conociendo rápidamente sus principales características. Son especialmente útiles en los recursos que no son textuales y en los que su contenido no puede ser indizado por sistemas automáticos, por ejemplo, los multimedia o un audio.

Pero los metadatos no sólo son descriptivos, también pueden ser administrativos y de estructura:

- *Metadatos descriptivos*: tienen propósito de descubrimiento (cómo se encuentra un recurso), identificación (cómo un recurso puede distinguirse de otro), y selección (cómo determinar que un recurso cubre una necesidad particular).
- *Metadatos administrativos*: es información que facilita la administración de los recursos. Incluyen información sobre cuándo y cómo fue creado el recurso, quién es el responsable del acceso o de la actualización del contenido y también se incluye información técnica, como la versión de software o el hardware necesario para ejecutar dicho recurso.
- *Metadatos estructurales*: sirven para identificar cada una de las partes que componen al recurso, definen la estructura que le da forma. Por ejemplo, un libro, que contiene capítulos y páginas, se puede etiquetar con metadatos que identifican cada parte y la relación que guardan entre ellas. Se usan especialmente para el procesamiento de la máquina y por software de presentación o estilos.

Un **ROA** (**R**epositorio de **O**bjetos de **A**prendizaje) es una colección de recursos (objetos o unidades de aprendizaje) que son accesibles a través de una red. El objetivo de un repositorio es facilitar la reutilización de recursos educativos, facilitando el acceso a los recursos almacenados en el mismo

Las funciones básicas que debe proveer un ROA (según la ADL, *Advanced Distributed Learning*), con el fin de lograr satisfacer tanto al usuario como la interoperabilidad con otros Sistemas, son [31]:

- *Buscar*. Es la habilidad para localizar un objeto de aprendizaje apropiado. Esto incluye la habilidad para su despliegue.
- *Recuperar*. Recibir un objeto de aprendizaje que ha sido pedido.
- *Enviar*. Entregar a un repositorio un objeto de aprendizaje para ser almacenado.
- *Almacenar*. Poner dentro de un registro de datos un objeto, con un identificador único que le permita ser localizado.
- *Colectar*. Obtener metadatos de los objetos de otros repositorios por búsquedas federadas.
- *Publicar*. Proveer metadatos a otros repositorios.

1.1. Análisis y objetivos del Proyecto

Independientemente del etiquetado utilizado para los metadatos asociados a los objetos de aprendizaje y de la plataforma (de software o hardware) sobre la que este implementado cada uno de los repositorios que van a formar parte de la federación, se quiere poder realizar búsquedas federadas y presentar al usuario la información, de forma que pueda acceder al objeto de aprendizaje y al conjunto de metadatos que lo describen.

Estamos en condiciones de determinar, como punto de partida, la siguiente proposición, que justifique el desarrollo de este trabajo:

La investigación objeto de esta tesis se justifica en la creación de una arquitectura de software para una federación de repositorios de objetos de aprendizaje, que permita localizar objetos de aprendizaje distribuidos en diferentes repositorios, independientemente de su ubicación física y de su tecnología de almacenamiento a través de Internet, de forma que haga que los sistemas en cuestión sean integrables e interoperables.

Un sistema construido en base a la arquitectura que vamos a proponer, y que se denominará Localizador de **O**bjetos de **A**prendizaje **D**istribuidos (**LOAD**), permitirá al usuario obtener objetos de aprendizaje de repositorios externos a partir de los criterios de búsqueda indicados.

Los objetivos del Proyecto son:

- Estudiar las características de los repositorios existentes en la actualidad y su adaptabilidad a las especificaciones y estándares
- Analizar el modo de comunicación entre los repositorios y la federación, evaluando ventajas y desventajas de las alternativas existentes, analizando los diferentes trabajos realizados en el área

- Dar una propuesta de una arquitectura que permita:
 - realizar búsquedas sobre los repositorios que integran la federación
 - sea abierta y fácil de adaptar a los sistemas actuales, es decir que permita la integración de dichos sistemas
- Implementar un prototipo para validar la arquitectura de la solución

Se lograron alcanzar los objetivos planteados al finalizar el Proyecto.

La comunicación de la federación con los repositorios se realiza a través de la API de consulta SQI (ver sección 2.3). Las operaciones provistas por esta API son invocadas a través de servicios Web. Para cada consulta sobre un repositorio se obtiene el conjunto de metadatos de los objetos de aprendizaje que cumplen con los parámetros de la consulta. Este conjunto de metadatos es representado en el estándar LOM en formato XML. Adicionalmente se podrá acceder a descargar el contenido educativo del objeto de aprendizaje devuelto.

Se adoptó como esquema de representación de los metadatos el estándar LOM, debido a que hay una marcada tendencia en el área a utilizar este esquema (por tratarse de un estándar) por los repositorios sobre los que se quiere trabajar. Igualmente la arquitectura es fácilmente extensible a otros esquemas como lo es Dublin Core.

Por otra parte, para la incorporación de un repositorio a la federación, basta con que el repositorio brinde acceso a través de la API SQI, accediendo a ésta mediante servicios Web, por lo tanto en este punto es fácilmente adaptable, independientemente de las heterogeneidades en cuanto a la plataforma sobre la que el repositorio este implementado y el lenguaje de consultas utilizado.

1.2. Estructura del documento

Este documento consta de siete capítulos y dos anexos, estructurándose como se describe a continuación.

En la **Sección 1**, *Introducción*, se incluye una introducción con la motivación y justificación de la investigación que se ha llevado a cabo, así como los objetivos y actividades realizadas durante el desarrollo del trabajo para alcanzar los objetivos planteados.

En la **Sección 2**, *Marco Conceptual*, desarrollamos un estudio en el área de los sistemas de aprendizaje, estudiando los estándares actuales dentro de la educación basada en Internet, ya que entendemos que cualquier propuesta que se haga hoy en día tiene que contemplar la interoperabilidad entre sistemas y la reutilización de los recursos educativos. A continuación se estudiarán las diferentes tecnologías que hacen posible la definición de la arquitectura, entre ellas se destacan, la API de acceso a los repositorios SQI, las arquitecturas orientadas a servicios y los servicios Web.

La **Sección 3**, *Relevamiento de Federaciones de Objetos de Aprendizaje*, se analizan los proyectos vinculados en el área, investigando las ventajas y desventajas de las propuestas.

En la **Sección 4**, *Arquitectura propuesta*, se expone la arquitectura propuesta, presentando las decisiones y composición estructural del sistema.

La **Sección 5**, *Implementación*, muestra las decisiones tomadas en la implementación del prototipo, indicando las tecnologías utilizadas, así como también la justificación en la elección de este tipo de tecnologías.

En la **Sección 6**, *Conclusiones y Trabajos Futuro*, se presenta un análisis y conclusiones sobre el proyecto, incluyendo posibilidades de mejora y extensiones a implementar sobre el prototipo.

La **Sección 7**, *Bibliografía y Referencias*, contiene las referencias bibliográficas.

2. Marco Conceptual

Como se mencionó en la Sección 1.1 el principal problema a resolver, es el de lograr la interoperabilidad entre los diferentes sistemas (repositorios de objetos de aprendizaje) y la federación. Esto motivo a realizar un estudio sobre las características principales de los mismos, los estándares vinculados en el área y las alternativas tecnológicas que hacen posible la interacción entre sistemas con estas características.

En esta Sección se presentan brevemente los conceptos base que se manejan en el resto del documento: Metadatos para *e-learning* y sus estándares, arquitectura SOA, servicios Web y SQI (Interfaz Simple de Consulta, del inglés “*Simple Query Interface*”) para los Repositorios de Objetos de Aprendizaje.

2.1. Repositorios y objetos de aprendizaje

Dada la variedad de las definiciones, así como la diversidad de recursos que pueden considerarse como OA; se considerará que cualquier recurso con una intención formativa, compuesto de uno o varios elementos digitales, descrito con metadatos, que pueda ser utilizado y reutilizados dentro de un entorno *e-learning* puede considerarse un OA.

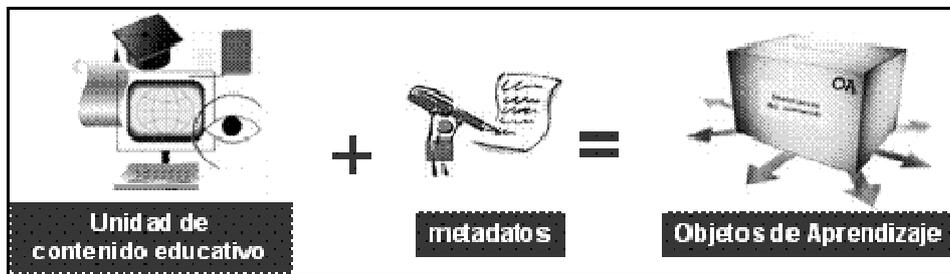


Figura 2– Componentes de los Objetos de Aprendizaje

Por lo tanto, el empleo de objetos de aprendizaje permite reutilizar los contenidos creados de una determinada experiencia educativa en contextos de aprendizaje diferentes. El proceso de construcción y distribución a los usuarios del material docente implica la creación, descubrimiento y agregación de unidades de aprendizaje simples en recursos educativos más complejos.

En el ámbito del *e-learning*, para la descripción de OA, se ha desarrollado el estándar, por parte de la IEEE, **LOM** (*Learning Object Metadata*) del que parten importantes iniciativas para la estandarización del *e-learning*. En LOM se especifica la sintaxis y la semántica de los atributos necesarios para describir los objetos de aprendizaje.

Con el uso de esquemas estándares de metadatos se busca, además de la organización, la reutilización de recursos y la interoperabilidad entre los sistemas involucrados con el uso de contenidos. Para hacer esto realmente posible es necesario que los metadatos estén representados a través de lenguajes abiertos, como XML (lenguaje de marcas extensible, del inglés “*Extensible Markup Language*”).

No se necesita un conocimiento previo de la estructura de la colección, la cual puede contener los propios recursos o únicamente los metadatos que los describen, junto con una referencia para su localización (según **IMS**, *Innovation Adoption Learning*) [7]. Los servicios que un repositorio de este tipo debe ofrecer al exterior, están relacionados con la búsqueda de objetos de aprendizaje a partir de metadatos, con el acceso a los objetos de aprendizaje localizados, o con el almacenamiento de objetos de aprendizaje en el repositorio.

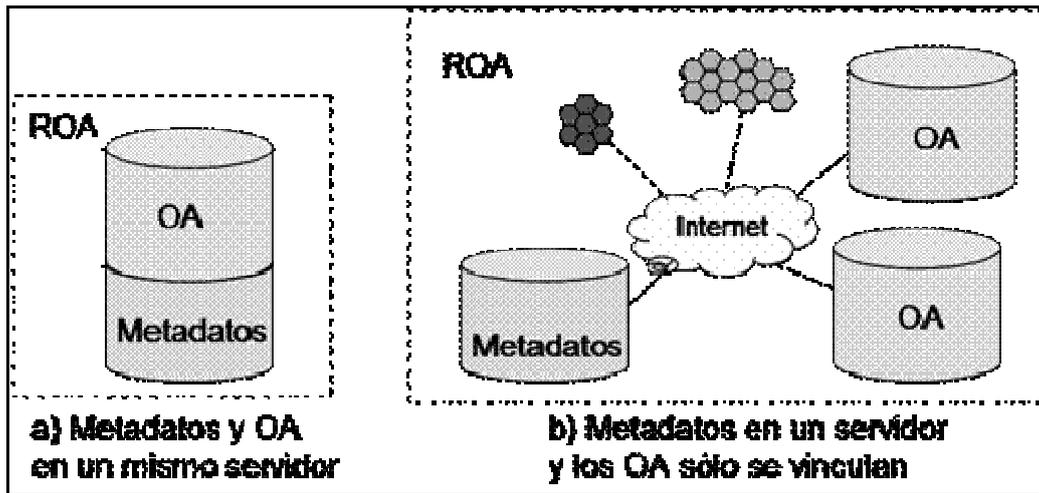


Figura 3 – Diferentes estructuras de repositorios

Como muestra la Figura 3, los tipos de ROA vienen dados por la forma en que estos almacenan los recursos, según Downes [15] existen dos tipos de repositorios:

- Los que contienen los OA y sus metadatos, en este caso los OA y sus descriptores se encuentran dentro de un mismo Sistema, e incluso dentro de un mismo Servidor. A estos se los conoce como Repositorios “centralizados”
- Los que mantienen por separado los OA y sus metadatos, en este caso el Repositorio almacena descriptores, los cuales tienen una referencia a la ubicación física del OA que se encuentra en otro Sistema, conocidos comúnmente como Repositorios “distribuidos”.

2.2. Estándares de Metadatos en *e-learning*

Con la especificación de estándares se busca conseguir un lenguaje común para facilitar la integración e interoperabilidad entre sistemas heterogéneos con pérdidas mínimas tanto de funcionalidad como de contenido. En conclusión, la estandarización de las tecnologías aplicadas al aprendizaje pretende posibilitar la reutilización de recursos educativos y la interoperabilidad entre sistemas de software heterogéneos.

Los estándares en el *e-learning* tienen como objetivos:

- *Accesibilidad*: Proporcionar acceso de contenido desde cualquier lugar a través de un navegador de Internet sin importar la plataforma o el contenido en sí mismo.
- *Interoperabilidad*: El contenido debería ser independiente de herramienta o plataforma, de tal manera de poder utilizar diferentes plataformas para acceder a un mismo contenido.
- *Adaptabilidad*: Los estándares se refieren al hecho de poder facilitar la adaptación o personalización del entorno de aprendizaje.
- *Re-usabilidad*: Solo el uso de estándares nos facilitará el diseñar contenidos que puedan ser utilizados una y otra vez en diferentes asignaturas, cursos o programas educativos.
- *Durabilidad*: El contenido debería poder utilizarse sin importar cambios en la tecnología base en el cual se elaboró. Esto sin necesidad de tener que re-codificar o re-compilar programas de software.

2.2.1. Tipos de estándares en *e-learning*

Los estándares que se desarrollan hoy en día para la industria del *e-learning* se pueden clasificar en los siguientes tipos [8]:

- Sobre el contenido del curso: estructura de los contenidos, empaquetamiento de contenidos, seguimiento de resultados.
- Sobre el alumno: almacenamiento e intercambio de información del alumno, habilidades del alumno, privacidad y seguridad.
- Sobre la interoperabilidad: integración de componentes del LMS (*Learning Management System*) e interoperabilidad entre múltiples LMS.

La Figura 4 ilustra los puntos marcados anteriormente:

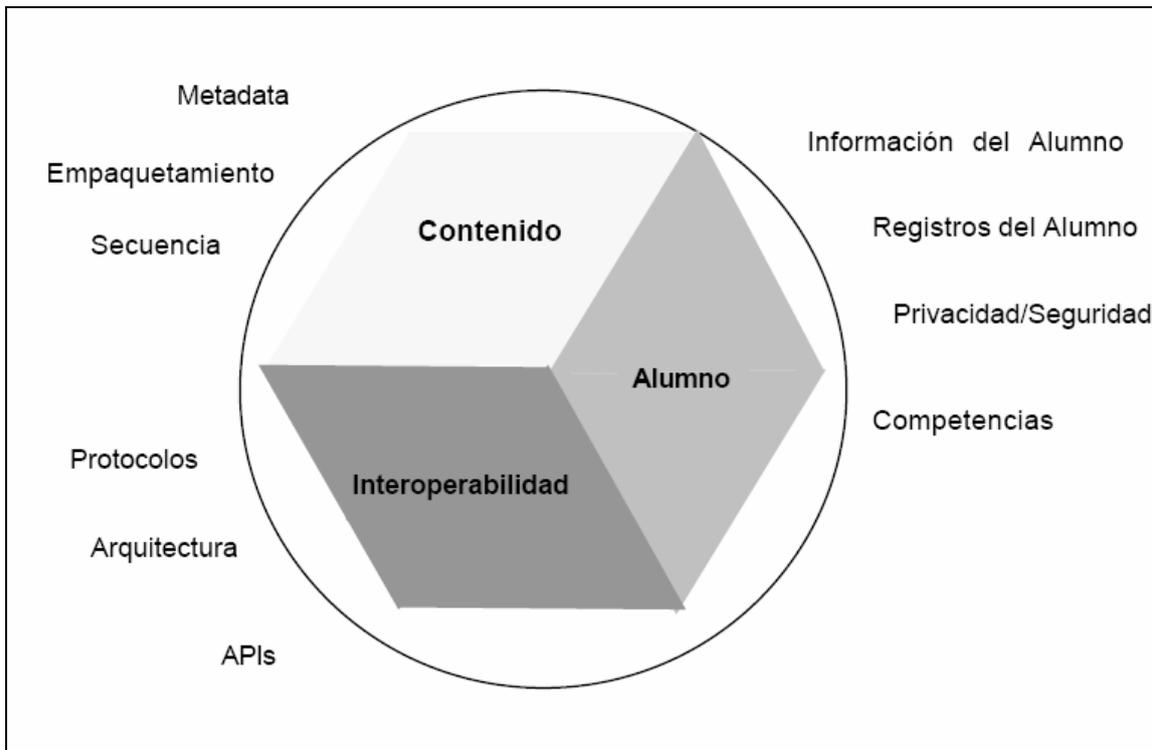


Figura 4 – Áreas afectadas por los estándares en e-learning [8]

Estrictamente hablando no existe un estándar *e-learning* hoy en día, sino lo que existe es una serie de grupos y organizaciones que desarrollan especificaciones, donde hasta el momento ninguna de estas especificaciones ha sido adoptada como estándar en la industria del *e-learning*.

Los estándares han iniciado el camino hacia una forma cómoda y viable de empaquetar los recursos y contenidos, tanto para los estudiantes que cambian de sistema, los docentes que utilizan en distintos contextos estos materiales y los desarrolladores que tienen que construir nuevas herramientas y mejorar las vigentes

2.2.2. Organizaciones de estandarización

Los organismos de estandarización más importantes en e-learning son:

- *AICC (Aviation Industry Computer Based Training Committee).*
- *IEEE (Institute of Electrical and Electronic Engineers)* con el grupo de trabajo LTSC (Learning Technology Standards Committee).
- *IMS (Instructional Management System) Global Learning Consortium.*
- *ADL (Advanced Distributed Learning).*
- *ARLADNE (Alliance of Remote Instructional Authoring and Distribution Networks for Europe).*
- *ISO (International Standards Organization)*, con el subcomité SC36 denominado “*Information Technology for Learning, Education and Training*”.

Estos abarcan las necesidades de definición en la industria en lo que refiere a: estructura de cursos, contenidos reutilizables, metadatos, arquitecturas de plataformas e intercambio de datos. Las mismas se encuentran trabajando en forma conjunta en la especificación y elaboración de estándares para el diseño de entornos tecnológicos en lo relativo al proceso de enseñanza a través de Internet.

Estas organizaciones se encuentran en este momento trabajando en la especificación de estándares para *e-learning* apuntando fundamentalmente a:

- **Repositorio de Contenidos:** Las organizaciones están orientando su esfuerzo en desarrollar estándares de contenidos *e-learning*. El principal objetivo es tener repositorios de objetos de aprendizaje reutilizables, de tal manera que puedan ser agrupados en objetos de aprendizaje adaptables y expedidos por cualquier plataforma *e-learning*. Sin embargo, uno de los mayores problemas al que se enfrenta hoy en día la industria del *e-learning* es la interoperabilidad entre distintos sistemas de aprendizaje que quieran compartir sus contenidos.
- **Internacionalización y Localización:** Los distintos grupos que están desarrollando especificaciones para *e-learning* participan de forma activa en todo el mundo, y cada día existe una mayor colaboración entre ellos. Esto genera dos desafíos: la creación de estándares “culturalmente” neutrales (internacionalización), y la adaptación de los estándares a las necesidades locales (localización).
- **Programas de certificación:** Existe un creciente énfasis en crear test de compatibilidad y programas de certificación. ADL está trabajando en un programa de certificación. Actualmente sólo existen los programas de certificación de AICC.
- **Arquitectura:** La industria del *e-learning* ha estado creciendo sin tener una clara visión de los componentes de un sistema de *e-learning* y de la forma en que interactúan. La necesidad de definir una arquitectura global es crítica para la evolución del desarrollo de estándares.

2.2.3. Metadatos

Como se mencionó en la Sección 2.2.1 existen diferentes tipos de estándares, en esta Sección nos enfocamos al estudio de los estándares “Sobre la interoperabilidad”, más precisamente los que han surgido vinculados a los metadatos que describen los objetos de aprendizaje: LOM, Dublin Core y SCORM. Para cada uno de ellos se presentan el conjunto de propiedades que los caracterizan y como se implementan.

2.2.3.1. LOM

En el año 2002 se emite el estándar por la IEEE que acredita al modelo de datos LOM [8] como el estándar de metadatos para describir el contenido de los objetos de aprendizaje. LOM especifica la semántica y la sintáctica de un conjunto mínimo de metadatos necesario para: identificar, administrar, localizar y evaluar un OA. Su propósito es facilitar a usuarios y a sistemas automáticos la tarea de buscar, compartir e intercambiar OA, permitiendo el desarrollo de catálogos que tienen en cuenta la diversidad cultural e idiomática de los contextos en los que se puedan utilizar los objetos y sus metadatos.

LOM se compone de 76 elementos (pudiendo ser extensible) organizando los metadatos en forma jerárquica con el objetivo de lograr una mejor organización y estructura. No es de comprensión trivial, por lo que las condiciones para llenarlos de forma adecuada deben estudiarse previamente. Para poder asignar valores, deben tenerse algunos conocimientos técnicos del recurso y conocimientos del campo pedagógico, por lo que es imprescindible la intervención humana (preferentemente especializada) y si bien existen proyectos donde se obtienen algunos datos de forma automática (EDUCA), ésta tarea tiene un nivel de dificultad alto.

La organización de los elementos viene dada por la agrupación en nueve categorías: general, ciclo de vida, metadatos, técnica, educativa, derechos, relación, anotación y clasificación. La Figura 5 [27] muestra lo antes mencionado, junto con las multiplicidades.

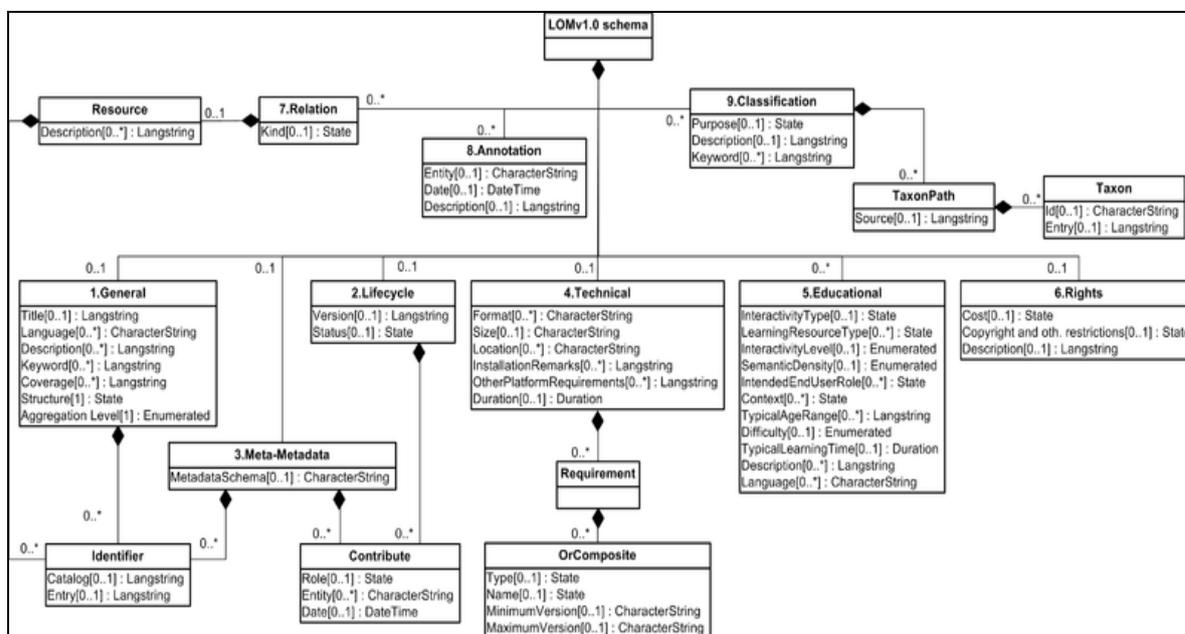


Figura 5 – Estructura general de LOM

La Tabla 1 muestra de forma más clara esta agrupación, mostrando entre “< >” cual es el nombre tal cual de la categoría para la implementación en XML, debajo de cada categoría los subelementos que la componen y en la columna derecha una breve descripción:

Categoría	Descripción
General <general> Identifier Title Catalog Entry Language Aggregation Level Description Keyword Coverage Structure Aggregation level	Información general que describe el objeto de aprendizaje como un todo.
Life Cycle <lifecycle> Version Status Contribute	Características relacionadas con la historia y el estado presente del objeto de aprendizaje y de aquéllos que han afectado a este objeto durante su evolución.
Meta-Metadata <metametadata> Identifier Contribute Metadata Schema Catalog Entry Language	Información sobre los mismos metadatos, no sobre el objeto de aprendizaje que se está describiendo.
Technical <technical> Format Size Location Requirement Installation Remarks Other Platform Requirements	Requisitos y características técnicas del objeto de aprendizaje.
Educational <educational> Interactivity Type Learning Resource Type Interactivity Level Semantic Density Intended End User Role Context Typical Age Range Difficulty Typical Learning Time Description Language	Condiciones del uso educativo del recurso.
Rights <rights> Cost Copyright and Other Restrictions Description	Condiciones de uso para la explotación del recurso.
Relation <relation> Kind Resource	Relación del recurso descrito con otros objetos de aprendizaje.
Annotation <annotation> Entity Date Description	Comentarios sobre el uso educativo del objeto de aprendizaje.

Classification <classification> Purpose Taxon Path Description Keyword	Descripción temática del recurso en algún sistema de clasificación.
---	---

Tabla 1 - Componentes de LOM

2.2.3.2. Dublin Core

El estándar Dublin Core [28] consta de 15 elementos, cuya semántica ha sido establecida mediante consenso por un grupo internacional de profesionales de distintas disciplinas, como biblioteconomía, informática, codificación de textos, museología y otros campos de conocimiento relacionados.

El formato Dublin Core permite que todos los campos sean repetibles, opcionales y puedan tener asociados cuantos enlaces se consideren necesarios.

En la Tabla 2 se muestran los elementos que hacen al estándar, agrupado por los elementos relacionados con el contenido del OA, los elementos que hacen a la propiedad intelectual del OA y los elementos relacionados con la versión del OA.

Categoría	Elemento	Descripción
Contenido	Cobertura	La característica de cobertura espacial y/o temporal del contenido intelectual del recurso.
	Descripción	Una descripción textual del recurso
	Tipo	La categoría del recurso
	Relación	Un identificador de un segundo recurso y su relación con el recurso actual
	Fuente	Secuencia de caracteres utilizado para identificar unívocamente un trabajo a partir del cual proviene el recurso actual.
	Materia	Los tópicos del recurso. Típicamente, expresará las claves o frases que describen el contenido
	Título	El nombre dado a un recurso
Propiedad intelectual	Colaborador	Una persona u organización que haya tenido una contribución intelectual significativa en la creación del recurso
	Editor	La entidad responsable de hacer que el recurso se encuentre disponible en la red en su formato actual
	Derechos	Una referencia (URL, por ejemplo) para una nota sobre derechos de autor

	Creador	La persona u organización responsable de la creación del contenido intelectual del recurso
Aplicación	Fecha	Una fecha en la que el recurso se puso a disposición del usuario en su forma actual
	Formato	El formato de datos de un recurso, usado para identificar el software y posiblemente, el hardware que se necesitaría para mostrar el recurso.
	Identificador	Secuencia de caracteres usados para identificar unívocamente un recurso.
	Lengua	Lengua/s del contenido intelectual del recursos

Tabla 2 – Componentes de Dublin Core

Es importante hacer notar, por lo visto anteriormente, que el estándar Dublin Core puede ser considerado como un “subconjunto de LOM” [28] o, lo que es lo mismo, que LOM es una extensión del estándar Dublin Core.

2.2.3.3. SCORM

El modelo SCORM [31] es un conjunto de estándares y especificaciones para compartir, reutilizar, importar y exportar OA. Los sistemas de gestión de contenidos en Web originales en sus orígenes usaban formatos propietarios para los contenidos que distribuían. Como resultado, no era posible el intercambio de tales contenidos. Con SCORM se hace posible el crear contenidos que puedan importarse dentro de sistemas de gestión de aprendizaje diferentes, siempre que estos soporten la norma SCORM.

Los principales requerimientos que el modelo SCORM trata de satisfacer son [29]:

- **Accesibilidad:** capacidad de acceder a los componentes de enseñanza desde un sitio distante a través de las tecnologías Web, así como distribuirlos a otros sitios.
- **Adaptabilidad:** capacidad de personalizar la formación en función de las necesidades de las personas y organizaciones.
- **Durabilidad:** capacidad de resistir a la evolución de la tecnología sin necesitar una reconcepción, una reconfiguración o una reescritura del código.
- **Interoperabilidad:** capacidad de utilizarse en otro emplazamiento y con otro conjunto de herramientas o sobre otra plataforma de componentes de enseñanza desarrolladas dentro de un sitio, con un cierto conjunto de herramientas o sobre una cierta plataforma. Existen numerosos niveles de interoperabilidad.
- **Reusabilidad:** flexibilidad que permite integrar componentes de enseñanza dentro de múltiples contextos y aplicaciones.

Las especificaciones de SCORM (especificación de ADL) están organizadas como “libros” separados (figura 6). SCORM ha sido dividido en cuatro libros:

- *Libro 1: Scorm Overview*. Contiene una descripción general de la iniciativa de ADL, un análisis de SCORM, y un resumen de las especificaciones técnicas contenidas en las siguientes secciones.
- *Libro 2: Scorm Content Aggregation Model (CAM)*. Contiene una guía para identificar y agregar recursos dentro de un contenido de aprendizaje. Este libro describe una nomenclatura para el contenido de aprendizaje. Define responsabilidades y requerimientos para construir agregaciones de contenidos, como pueden ser cursos, lecciones, módulos, etc. Contiene información sobre la creación de contenidos aplicando metadatos y el secuenciamiento y navegación en el contexto del empaquetamiento de los contenidos (SCORM Content Packaging). Este libro está fuertemente relacionado con el libro 3 (Scorm Run-Time Environment). Los metadatos SCORM describen los diferentes componentes del Scorm *Content Aggregation Model* (Agregaciones de contenidos, Actividades, SCO's y Assets). Los metadatos es una forma de etiquetar esos componentes para facilitar su búsqueda.
- *Libro 3: Scorm Run-Time Environment (RTE)*. Incluye una guía para lanzar contenidos y hacerles un seguimiento en un ambiente basado en Web de un entorno de ejecución que incluye: un protocolo específico para la ejecución de contenidos Web, un API entre el contenido y el LMS y un modelo de datos que define el flujo de datos intercambiado entre el entorno LMS y el contenido que se ejecuta en el entorno de ejecución.
- *Libro 4: Scorm Sequencing and Navigaton (SN)*. Este libro describe cómo el contenido debe ser secuenciado a través del sistema junto con los eventos de navegación. El contenido puede ser descrito por un conjunto de actividades predefinidas, definidas durante el diseño de los contenidos. Este libro describe como un LMS de SCORM interpreta las reglas de secuenciamiento expresadas en un entorno de desarrollo y sus efectos en el *Run-Time Environment*. Describe las ramificaciones y el camino que siguen las actividades de aprendizaje, representadas mediante un Árbol de Actividad, basadas en los resultados de las interacciones del estudiante con los objetos de contenido y una estrategia de secuenciamiento. Un Árbol de Actividad es una estructura conceptual de actividades manejadas por el LMS propias de cada alumno. En SCORM, una actividad de aprendizaje hace referencia a objetos de contenidos que cursa el estudiante. Está basado en la especificación propuesta para el secuenciamiento de contenidos de IMS.

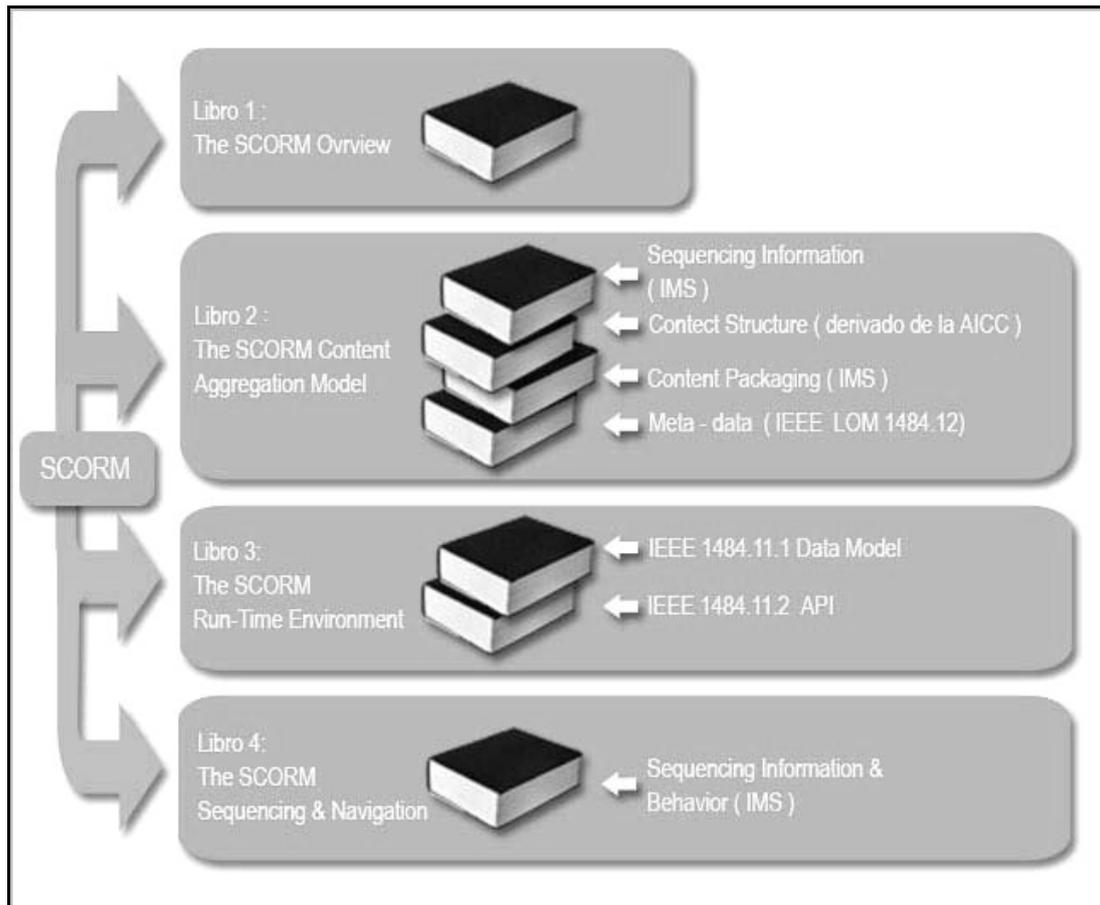


Figura 6 – Organización de las especificaciones SCORM [30]

2.3. SQI (*Simple Query Interface*)

SQI es una API para el establecimiento de sesión y realización de consultas síncronas y asíncronas que define los servicios que un repositorio debe de tener disponibles para recibir y responder consultas de otros repositorios.

Es parte de la arquitectura para la interoperabilidad de repositorios educativos **LORI** (en inglés, *Learning Object Review Instrument*), la cual define los servicios necesarios para permitir la interoperabilidad entre estos repositorios [18]. SQI sólo se encarga del envío y recepción de consultas, no teniendo en cuenta en ningún momento la estructura de las mismas, lo que lo hace sencillo de implementar en una gran variedad de sistemas heterogéneos.

Los servicios brindados incluyen servicios básicos como por ejemplo servicios de autenticación, gestión de la sesión y servicios de aplicación como gestión de las consultas.

Las ventajas de SQI vienen asociadas a que es neutral respecto al lenguaje de consultas utilizado en el manejador de BD y en el modelo semántico seguido por el repositorio.

La Figura 7 [18] ilustra lo antes detallado mostrando la comunicación entre dos repositorios, uno origen y otro destino:

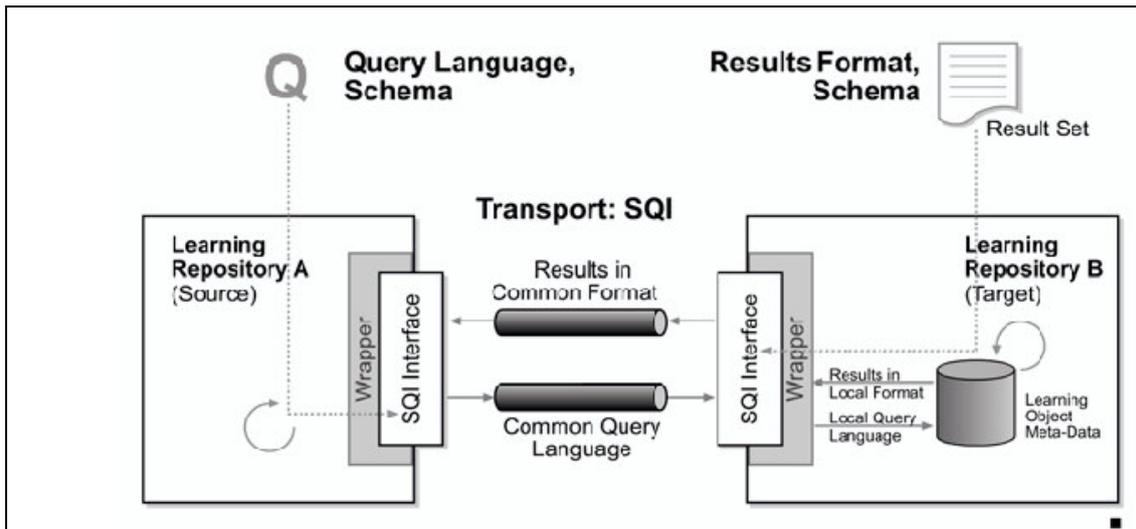


Figura 7 – Modelo de comunicación SQI [13]

2.3.1. Lenguaje de consulta y formato de resultados

Para aplicar plenamente la funcionalidad de consultas en la API SQI se debe establecer acuerdos acerca de:

- El conjunto de atributos y vocabularios que pueden ser usados en la consulta.
- El lenguaje de consultas y su representación.
- La representación de la lista de objetos de aprendizaje que satisfagan la consulta.
- La representación de los metadatos de esos objetos de aprendizaje.

2.3.2. Síncrono y Asíncrono

SQI presenta dos escenarios diferentes: en el escenario síncrono el destino devuelve los resultados a la fuente, por lo tanto la recuperación de los resultados es iniciada por la fuente. El proceso es el siguiente, la fuente realiza la consulta y a través de otros métodos accede después a los resultados.

En el escenario asíncrono la transmisión de resultados es responsabilidad del destino, es decir una vez iniciada la consulta y a medida que se obtienen los resultados, éstos son transmitidos a la fuente. Para hacer posible este escenario la fuente tiene que tener la capacidad de identificar una consulta enviada a un destino particular, por ejemplo si la misma consulta fue enviada a diferentes destinos. También, la fuente, debe de implementar un punto de acceso para la recepción de los resultados.

Tanto en el escenario síncrono como asíncrono es importante hacer notar que es posible realizar múltiples consultas por sesión, inclusive simultáneamente.

2.3.3. Gestión de sesión

La sesión hace posibles establecer un vínculo directo entre la fuente y el destino. Cada sesión tiene un identificador asociado (ID sesión) el cual sirve para identificar la fuente/destino de las consultas.

Por lo tanto SQI está basado en un manejo simple del concepto de sesión. La sesión debe ser establecida antes que se establezca una nueva comunicación.

Por ejemplo en el caso de tenerse un escenario síncrono el origen establece durante un período una sesión en el destino usando el ID de sesión obtenido para identificarse a sí mismo durante toda la comunicación.

La sesión es válida hasta el momento en que es destruida, por lo tanto sigue estando activa después que una consulta es ejecutada. Por otra parte existe un período de sesiones, o sea si durante un tiempo determinado no se lleva a cabo una comunicación la sesión es eliminada.

2.3.4. *Stateless/Stateful*

Son atributos en el caso de repositorios diseñados para realizar un seguimiento de acontecimientos en una secuencia de interacciones. Stateless tiene como significado que no existe un registro de las anteriores interacciones y que cada solicitud debe resolverse únicamente con la información que viene con ella. En el caso de Stateful el repositorio destino realiza un seguimiento del estado de interacción, por ejemplo, a través de almacenar los resultados obtenidos de una consulta en un cache. SQI permite la opción de optar por los diferentes casos.

2.3.5. Principio de separación Comando-Consulta

SQI está diseñado siguiendo este principio, que establece que cada método que realiza es un comando que realiza una acción o una consulta que devuelve datos, pero no ambos. Con esto se logran diseños de interfaces más claros y comprensibles.

2.3.6. Métodos de SQI

A continuación se detallan los distintos métodos que proveen SQI y una descripción desde la perspectiva de trabajo. En primer lugar la fuente debe establecer una conexión con el destino, por ej. usando “createAnonymousSession”. Una vez establecida la conexión la interfaz de consulta en el destino queda a la espera de alguna solicitud de búsqueda. También la posibilidad de configuraciones en el destino a través de otros métodos, por ej. parámetros de consulta como:

- lenguaje de consultas (setQueryLanguage),
- el número de resultados obtenidos dentro de un conjunto de resultados (setResultsSetSize),
- el número máximo de los resultados de una consulta (setMaxQueryResults),
- la duración máxima de ejecución de consultas (setMaxDuration),
- y el formato de resultados (setResultsFormat).

Todos los parámetros establecidos a través de estos métodos continúan siendo válidos mientras dura el período de sesión. En caso de no realizarse modificaciones mantienen los valores por defecto.

Métodos para la gestión de sesión	<i>createSession</i>
	<i>createAnonymousSession</i>
	<i>destroySession</i>
Métodos configuración de parámetros de consulta	<i>setResultsFormat</i>
	<i>setMaxQueryResults</i>
	<i>setMaxDuration</i>

Métodos interface de consulta Síncrona	<i>setResultsSetSize</i>
	<i>synchronousQuery</i>
	<i>getTotalResultsCount</i>
Métodos interface de consulta Asíncrona	<i>asynchronousQuery</i>
	<i>setSourceLocation</i>

Tabla 3 – Métodos de la API SQI

2.4. Servicios Web y Arquitectura SOA

Internet introduce un nuevo entorno donde el software se puede ofrecer y acceder como servicio. Los servicios Web proporcionan la plataforma ideal para conseguir la completa integración de los procesos de negocio de una determinada organización con el resto de implicados en dichos procesos (clientes, proveedores, etc.) [23].

Los servicios Web, se proponen como una alternativa para facilitar la intercomunicación entre diferentes arquitecturas de componentes, ofreciendo una visión de dichas arquitecturas, basada en servicios, totalmente compatible con Internet.

La aparición de los servicios Web, y de las Arquitecturas Orientadas a Servicios (SOA) supone el establecimiento de nuevos mecanismos de comunicación B2B (*Business to Business*), B2C (*Business to Consumer*), B2E (*Business to Employee*). El organismo encargado de definir estos estándares, y asegurar este comportamiento, es el WS-I (*Web Services Interoperability Organization*). Gracias a este organismo, va a ser posible que sistemas desarrollados en diferentes plataformas y diferentes lenguajes de programación, puedan interactuar.

La Arquitectura Orientada a Servicios (SOA), que se muestra detalladamente en un siguiente punto, pretende extender la idea de servicio Web, de forma que una invocación, de manera totalmente transparente al usuario, implique la ejecución de más de un servicio Web.

El modelo de servicios Web sirve de gran apoyo en el desarrollo de aplicaciones distribuidas. Un ejemplo podría ser el caso de una agencia de viajes que tiene asociado su sistema de reservas on-line con el sistema de aerolíneas, reserva de hoteles y alquiler de coches, cada uno de estos subsistemas independientes entre sí; de tal manera que un viajero puede al mismo tiempo hacer una reserva de un vuelo, una habitación de hotel y un coche de alquiler, todo esto a través servicios Web intercomunicados entre sí para conseguir dicha finalidad, constituyendo lo que se conoce como arquitectura orientada a servicios, no poniendo ninguna limitante en cuanto a la plataforma, ubicación y lenguaje de implementación sobre los que están desarrollados cada uno de los subsistemas.

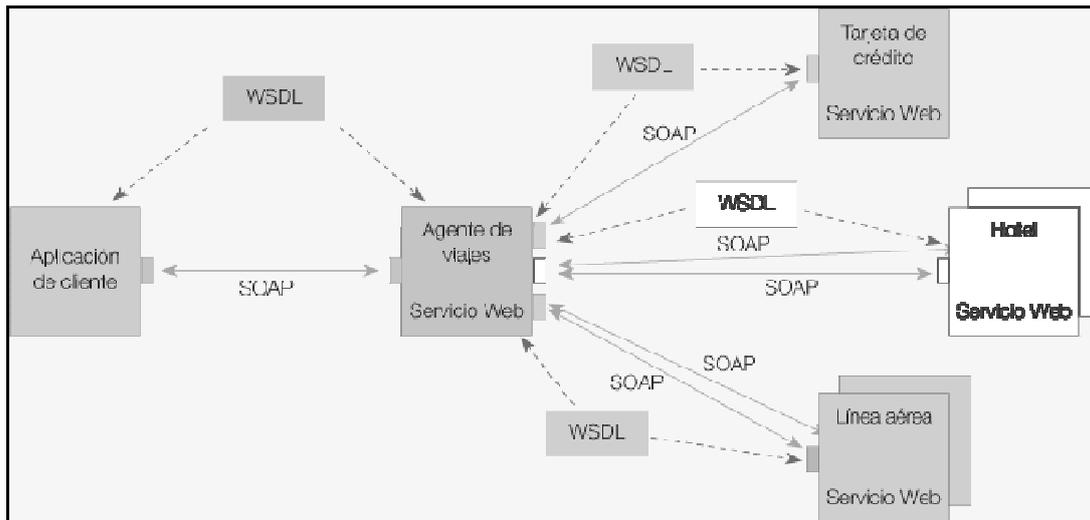


Figura 8 - Funcionamiento de los servicios Web [25]

La arquitectura de los servicios Web permite que ciertos servicios sean dinámicamente descritos, publicados, descubiertos e invocados en un ambiente de computación distribuida, haciendo uso de los estándares WSDL, UDDI, SOAP y XML respectivamente.

2.4.1. Descripción general de los Servicios Web

Como ya se ha indicado, estos servicios se basan en un conjunto de estándares (WSDL, UDDI, XML y SOAP, que serán comentados más adelante) que permiten a los desarrolladores implementar aplicaciones distribuidas, utilizando herramientas muy distintas para crear aplicaciones que utilizan una combinación de módulos de software que son llamados desde diversos sistemas distribuidos en regiones geográficas distintas.

Los servicios Web son aplicaciones auto-contenidas y modulares que pueden ser [24]:

- Descritas mediante un lenguaje de descripción de servicio, como el lenguaje WSDL (*Web Service Description Language*).
- Publicadas, al incluir las descripciones y políticas de uso en algún registro conocido, utilizando el método de registro UDDI (*Universal Description, Discovery and Integration*).
- Encontradas, también utilizando el estándar UDDI, al enviar peticiones al registro y recibir los detalles necesarios para la localización y enlace (del inglés, *binding*) sobre aquel servicio que se ajusta a los parámetros de la búsqueda.
- Asociadas, al utilizar la información contenida en la descripción del servicio para crear una instancia de servicio disponible (o Proxy).
- Invocadas sobre la red, al utilizar la información contenida en los detalles de enlace de la descripción del servicio; en un documento WSDL. Durante la invocación, como veremos más adelante haremos uso del protocolo SOAP.
- Compuestas con otros servicios para integrar servicios y aplicaciones nuevas, en lo que constituirá la base de SOA, que será explicada en detalle en un apartado posterior.

Todos los estándares comentados se basan en XML: los documentos WSDL, son documentos XML; el protocolo SOAP, que permite la comunicación entre servicios, internamente trata información XML en sus dos variantes, por un lado la mensajería

SOAP, tratando información XML explícita y por otro lado RPC, que la trata, pero de una manera implícita.

En cuanto a los componentes de una arquitectura orientada a servicios Web, podemos hablar de [25]:

- **Servicio:** La aplicación es publicada para que sea utilizada por todos aquellos solicitantes que cumplan los requisitos especificados por el proveedor de servicios. Evidentemente la implementación se realizará sobre una plataforma accesible en la red. El servicio se describe a través del lenguaje de descripción de servicios, WSDL. Tanto la descripción como las políticas de uso han sido publicadas anteriormente en un registro.
- **Proveedor de Servicio:** Desde el punto de vista comercial, es quien presta el servicio. Desde el punto de vista de la arquitectura, es la plataforma que provee el servicio.
- **Registro de Servicios:** Es un repositorio de descripciones, donde los proveedores publican sus servicios y la forma de accederlos. Permitirá además a los solicitantes realizar distintos tipos de búsquedas, obteniendo de éstas, los detalles necesarios para poder localizarlos y utilizarlos.
- **Solicitante de servicios:** Desde el punto de vista comercial, la empresa que requiere cierto servicio. Desde el punto de vista de la arquitectura, la aplicación cliente que busca e invoca un servicio.

Las operaciones que se pueden realizar con los servicios Web, son las siguientes:

- *Publicar/Cancelar:* Un proveedor de servicios podría publicar un determinado servicio comercial (*e-business*) a uno o más registros de servicios, además de, evidentemente, cancelar dicha publicación en un momento dado.
- *Búscar:* Los solicitantes de servicios (clientes) podrán interactuar con uno o más registros para descubrir un conjunto de servicios comerciales con los que puedan interactuar para encontrar una solución a sus problemas.
- *Enlazar (Bind):* Los solicitantes de servicios negocian con los proveedores la forma de acceder e invocar a sus servicios comerciales (*e-business*).

Entre las razones por las cuales los servicios Web jugarán un rol principal en la siguiente generación de sistemas distribuidos, están las siguientes:

- **Interoperabilidad:** Cualquier servicio Web puede interactuar con cualquier otro servicio. El protocolo estándar SOAP permite que cualquier servicio pueda ser ofrecido o utilizado independientemente del lenguaje o ambiente en que se haya desarrollado.
- **Omnipresencia:** Los servicios Web se comunican utilizando HTTP y XML. Cualquier dispositivo que trabaje con éstas tecnologías puede ser tanto cliente y acceder a los servicios Web. Por ejemplo, una máquina de venta de refrescos, incluso podría comunicarse vía inalámbrica con el servicio Web de un proveedor local y ordenar un pedido de suministro.
- **Barrera mínima de participación:** Los conceptos que hay detrás de los servicios de Web son fáciles de comprender y se ofrecen infinidad de herramientas de desarrollo (ToolKits); como por ejemplo las ofrecidas por IBM, Sun Microsystems, Apache,

IIS o Systinet. Todas ellas permiten a los desarrolladores crear e implementar rápidamente servicios Web.

- Apoyo de las Industrias: Las principales compañías apoyan el protocolo SOAP y la tecnología derivada de los servicios Web.

2.4.2. Estándares y tecnologías

Los servicios Web basan todo su funcionamiento y características en los siguientes estándares y protocolos [25]:

- ✓ XML (*eXtensible Markup Language*) ha revolucionado la forma en que se estructura, describe e intercambia información. Independientemente de las múltiples formas en que se utiliza hoy en día XML, todas las tecnologías de servicios Web se basan en XML. Se trata del estándar central de esta arquitectura, sobre el cual se apoyan el resto. El diseño de XML deriva de dos fuentes principales: SGML (*Standard Generalized Markup Language*) y de HTML (*HyperText Markup Language*).
- ✓ UDDI (*Universal Description, Discovery and Integration*), es un directorio que contiene un registro/repositorio de descripciones de servicios Web. Este estándar permite a las empresas registrarse en un tipo de directorio de Internet que les ayuda a anunciar sus servicios, de tal forma que el resto de compañías puedan localizar sus servicios y realizar transacciones en la Web. El proceso de registro y consultas se realiza utilizando mecanismos basados en XML y HTTP(S). Por lo tanto, la especificación UDDI tiene dos objetivos esenciales, en primer lugar servir de soporte a los desarrolladores para encontrar información sobre servicios Web y poder construir clientes; y por otro lado, facilitar el enlace dinámico de servicios Web, permitiendo consultar referencias y acceder a servicios de interés.
- ✓ SOAP (*Simple Object Access Protocol*) es un estándar del Consorcio WWW que define un protocolo que da soporte a la interacción (datos + funcionalidad) entre aplicaciones en entornos distribuidos y heterogéneos, es interoperable (neutral a la plataforma, lenguajes de programación, independiente del hardware y protocolos). Funciona sobre la infraestructura (estándares) existentes en Internet. SOAP define cómo organizar la información XML de una manera estructurada y tipada para intercambiarla entre los distintos sistemas. El protocolo SOAP simplifica el acceso a los objetos, permitiendo a las aplicaciones invocar métodos de objetos o funciones, que residen en sistemas remotos.
- ✓ WSDL (*Web Service Description Language*) creado originalmente por IBM, Microsoft y Ariba. Tiene un rol y un propósito similar al de los IDL (*Interface Definition Language*) de las plataformas *middleware*. Un archivo WSDL es un documento XML que describe los servicios Web, en particular sus interfaces. Como característica que lo diferencia de los IDL, es que WSDL debe definir los mecanismos de acceso (protocolos) a los servicios Web. Otra característica diferenciadora es la necesidad de definir (en la especificación) la localización del servicio (puntos finales). La separación de interfaces y enlaces de protocolos, y la necesidad de incluir información de localización permite la definición de especificaciones modulares. WSDL permite definir interfaces más complejas y expresivas; permitiendo definiciones de interacciones asíncronas y diferentes paradigmas de interacción, y la posibilidad de combinar o agrupar operaciones.

2.4.3. Arquitectura Orientada a Servicios (SOA)

Los sistemas de información son cada vez más complejos y se necesitan sistemas más exigentes que requieren cooperación; y cada vez son menos comunes las soluciones centralizadas y aparecen nuevos métodos de interacción (comunicaciones) [25]. El objetivo de las organizaciones es reducir los costes y maximizar la utilización de la tecnología existente; al mismo tiempo las empresas intentan ser más competitivas y avanzar en sus prioridades estratégicas.

Existen dos líneas para conseguir estos propósitos: La heterogeneidad y el cambio. Muchas empresas tienen diferentes sistemas, aplicaciones y arquitecturas de diferentes tecnologías y algunas bastante antiguas. Integrar productos de diferentes proveedores y de diferentes tecnologías es el objetivo, por eso cada vez más las aplicaciones están orientadas al servicio.

Durante años la industria de la tecnología ha estado luchando para solucionar problemas como la heterogeneidad, la interoperabilidad (para que la solución escogida o la aplicación construida sean independientes de la plataforma, tecnología utilizada y ubicación).

La Arquitectura Orientada a Servicio presenta una ventaja para la construcción de sistemas distribuidos, ya que contempla, la funcionalidad de las aplicaciones como servicios que pueden utilizar otras aplicaciones y otros servicios.

Una Arquitectura Orientada a Servicio está compuesta por elementos funcionales y elementos relacionados con la calidad de servicio, que se describen a continuación.

Aspectos funcionales [25]:

- ✓ Transporte: Es el mecanismo usado para llevar peticiones de servicios desde el consumidor al proveedor del servicio, y las respuestas desde el proveedor del servicio al consumidor del servicio.
- ✓ Protocolo de comunicación del servicio: Es el mecanismo de comunicación establecido entre el proveedor del servicio y el consumidor del servicio.
- ✓ Descripción del servicio: Es el esquema establecido para describir qué es el servicio, cómo debe invocarse y que datos son requeridos para la invocación.
- ✓ Servicio: Describe un servicio que está disponible para utilizarse.
- ✓ Proceso de negocio: Es una colección de servicios, invocados de una manera particular, en una determinada secuencia y con unas reglas particulares para llevar a cabo la funcionalidad de negocio requerida. Un proceso de negocio puede estar compuesto por servicios de diferente naturaleza e incluso en distintas localizaciones.
- ✓ Registro de Servicio: Es el repositorio de servicios y las descripciones que son usados por los proveedores de servicio para publicarlos, y para que los consumidores del servicio puedan invocarlos. El registro del servicio puede aportar la funcionalidad a los servicios que necesiten un repositorio centralizado.

Aspectos de la calidad del servicio:

- ✓ Política: Es un conjunto de condiciones o reglas sobre las cuales un proveedor del servicio hace un servicio disponible a los consumidores.
- ✓ Seguridad: Es el conjunto de reglas que pueden ser aplicadas para la identificación, autorización y el control de acceso a los consumidores de servicios.

- ✓ **Transacción:** Es el conjunto de atributos que pueden ser aplicados a un grupo de servicios para conseguir un resultado consistente. Por ejemplo si un grupo de tres servicios tienen que terminar para completar la función, todos tienen que estar completados y haber terminado su ejecución.
- ✓ **Gestión:** Es el conjunto de atributos que pueden ser aplicados para manejar a los proveedores del servicio o a los consumidores.

Roles y operaciones

Los roles de una Arquitectura Orientada a Servicio son:

- ✓ **Consumidor de servicio:** El consumidor de servicio es una aplicación, un módulo de software u otro servicio que requiere un servicio. Inicia la búsqueda en el registro de servicio, enlaza con el servicio a través del transporte y ejecuta la función del servicio de acuerdo con las reglas establecidas.
- ✓ **Proveedor de servicios:** El proveedor de servicios es una entidad que se puede acceder a través de la red y que acepta y ejecuta peticiones de los consumidores. Publica las interfaces de los servicios en el registro de servicios para que los consumidores puedan descubrirlos y puedan acceder a ellos.
- ✓ **Registro de servicios:** Un registro de servicios es el que permite que los servicios puedan ser descubiertos. Contiene un repositorio con los registros que están disponibles y permite la búsqueda de los proveedores de los servicios a través de las interfaces que han sido establecidas y que son de interés para los consumidores.

Las operaciones dentro de una arquitectura orientada a servicios son:

- ✓ **Publicación:** Para que los servicios puedan estar accesibles, un servicio tiene que tener una descripción, que debe ser publicada para que pueda ser descubierta e invocada por un consumidor de servicio.
- ✓ **Localización:** Un consumidor del servicio puede localizar un servicio realizando una búsqueda sobre el registro de servicios que cumpla algún criterio.
- ✓ **Enlazar e invocar:** Después de recoger la descripción del servicio, el consumidor del servicio puede invocar el servicio de acuerdo con la información de la propia descripción del servicio.

Características de una Arquitectura Orientada a Servicios

Para que el funcionamiento de una arquitectura orientada a servicio sea dinámico, tiene que cumplir las siguientes características:

- Los servicios tienen que ser modulares
- Los servicios tienen que soportar la interoperabilidad
- Los servicios tienen que tener la descripción perfectamente establecida
- Los servicios tienen que ser transparentes a la localización
- Los servicios tienen que ser independientes del lenguaje de implementación
- Los servicios tienen que ser transparentes al protocolo de comunicación
- Los servicios tienen que ser independientes del Sistema Operativo y del hardware utilizado

2.4.4. Servicios Web y Arquitectura Orientada a Servicios

Como ya se vio en detalle puntos anteriores los servicios Web y las tecnologías que les ofrecen soporte; en este punto se justifica porque los servicios Web pueden llegar a ser una parte fundamental de una Arquitectura Orientada a Servicio [25].

Dentro de un proceso de negocio puede ser necesario utilizar funcionalidades de distintos sistemas y distintas localizaciones; para completar estas funcionalidades se utilizan los Servicios Web, que también tienen que ser identificados durante el proceso de análisis de una arquitectura orientada a servicio. Cada servicio tiene que estar bien definido mediante una interfaz (WSDL) para que pueda ser publicado, localizado e invocado. Dependiendo del proceso de negocio, el servicio puede ser publicado para que otras empresas puedan utilizarlo, o internamente para ser utilizado en los procesos de negocio internos de una determinada compañía.

Los servicios Web son una tecnología altamente adaptable a las necesidades de implementación de una Arquitectura Orientada a Servicio. En esencia, los servicios Web son la implementación de una especificación bien definida de una funcionalidad, es decir, son aplicaciones modulares que aportan una lógica del proceso de negocio como servicio que puede ser publicado, localizado e invocado en Internet [IBM, 2006]. Basados en los estándares XML [W3C, 1998], los servicios Web pueden ser desarrollados usando cualquier lenguaje de programación, cualquier protocolo y cualquier plataforma. Los Servicios Web pueden ser localizados y utilizados en cualquier momento, desde cualquier localización y usando cualquier protocolo y plataforma.

Pero es importante remarcar que los servicios Web no son la única tecnología que es usada para implementar una Arquitectura Orientada a Servicio. Existen ejemplos de organizaciones que utilizan con éxito una Arquitectura Orientada a Servicios donde utilizan, además de los servicios Web otras tecnologías de intercambio de mensajes y de acceso a funciones remotas haciendo uso del estándar XML, como puede ser el protocolo XML RPC [Winer, 1999], que funciona exactamente igual que el protocolo RPC (*Remote Procedure Call*), a través de un túnel HTTP.

3. Relevamiento de Federaciones

En la actualidad existen dos elementos clave para el desarrollo del *e-learning*: la reutilización de objetos de aprendizaje y la interoperabilidad entre los repositorios de objetos de aprendizaje. Para conseguir la completa reutilización de un objeto de aprendizaje se debe, en primer lugar, desarrollarlo conforme a estándares de *e-learning*, como los establecidos por IMS, ADL (SCORM) o IEEE (LOM) explicados en la Sección 2. De esta forma se asegura que, a través de su descripción mediante metadatos, pueda ser integrado en cualquier plataforma de *e-learning* compatible con estos estándares. En segundo lugar, se ha de publicar en un repositorio que garantice su localización automática por parte de los diferentes tipos de usuarios que puedan verse implicados en un proceso educativo. Pero para que esta reutilización sea completa, estos recursos se deben poder descubrir desde otros repositorios o plataformas de formación distribuidos a través de Internet.

En este sentido existen diferentes alternativas que siguen esta dirección, es decir repositorios de objetos de aprendizaje que tratan de compartir sus recursos. Estos repositorios tienen la misión de servir de punto de encuentro de los creadores de contenidos donde intercambiar los contenidos creados y así hacer que sean reutilizables.

Generalmente estos sistemas suelen ser gratuitos, tanto en su uso como los contenidos que almacenan; por lo tanto, no se puede garantizar la calidad del material docente que almacenan ni en sus contenidos ni en los metadatos que los describen. Por otro lado, estos repositorios no llegan a ser útiles si no utilizan metadatos para la clasificación de sus objetos de aprendizaje- Así como también muchas de las especificaciones de metadatos que existen hoy en día no ayudan demasiado a simplificar la búsqueda y reutilización, debido a su falta de completitud y simplicidad.

En este capítulo se presentan algunos de las federaciones que ofrecen sus contenidos en Internet de manera de comprender mejor sus propiedades y detectar sus carencias, pudiendo así obtener una visión crítica hacia cada uno de ellos.

La elección de las federaciones a presentar en este documento se basó en su riqueza de metadatos y en el material que hubiera disponible para poder llevar a cabo el análisis propuesto. A partir de este se realiza el planteo de la arquitectura, donde las decisiones de diseño van a estar influenciadas por las conclusiones de esta sección.

3.1. Estándar IMS DRI

La IMS DRIWG (*IMS Digital Repository Interoperability Working Group*) [7] es una especificación de normas y recomendaciones, que facilita un esquema funcional de la arquitectura del sistema y un modelo de referencia completo para la interoperabilidad de repositorios. Permite a los repositorios interactuar entre sí, ignorando su arquitectura interna, para lograr esto el esquema de funcionamiento propuesto es el que se detalla en el recuadro de la Figura 9:

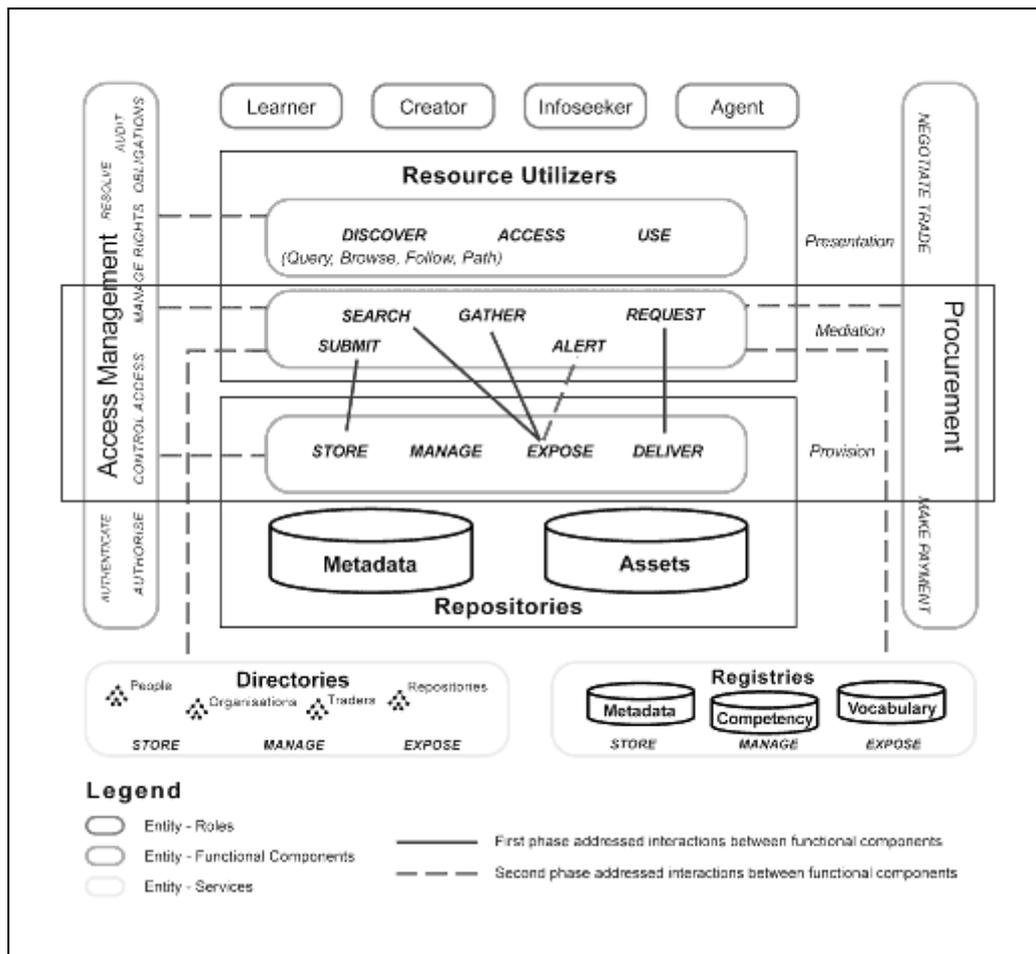


Figura 9 – Esquema de funcionamiento de la IMS DRI [7]

En el esquema propuesto se puede observar que la gestión del repositorio se basa en un conjunto de funciones, que la especificación agrupa en pares de funciones, las cuales se describen en la tabla siguiente:

Pares de Funciones	Funcionalidad
Buscar / Exponer	El usuario especifica los atributos que servirán para discriminar sobre los metadatos de los objetos de aprendizaje. La respuesta es un conjunto de metadatos.
Enviar / Almacenar	Esta funcionalidad hace referencia a la forma de almacenar un objeto en un repositorio y la forma que tomará una vez almacenado para hacer posible su recuperación. El lugar desde el cual se coge el objeto para su almacenamiento puede ser otro repositorio, un sistema enseñanza, el disco duro del desarrollador, o cualquier punto de la red.
Alertar / Exponer	La especificación contempla esta funcionalidad como un posible componente

	de un repositorio digital o un servicio intermedio encargado de mandar correos electrónicos.
Colectar/ Exponer	Corresponde a la activación periódica del mecanismo de búsqueda. Esta funcionalidad proporciona la forma de escribir los meta datos que van a servir para las búsquedas, la forma de agruparlos para facilitar los sondeos futuros y la manera en que se tienen que agregar para formar nuevos repositorios (estos almacenes estarán disponibles para las funciones de búsqueda y alerta). Esta funcionalidad interactúa con el repositorio de dos maneras diferentes. La primera consiste en solicitar metadatos del repositorio, mientras que en la segunda ofrece al almacén meta datos para que sean almacenados.
Solicitar / Entregar	La función Solicitar es la petición de acceso a un recurso que realiza un usuario del sistema una vez lo ha localizado gracias a los meta datos que lleva asociados. Entregar se refiere a la respuesta que le da el repositorio, que le otorga o le niega el acceso al recurso.

Tabla 4 – Conjunto de funciones de la IMS DRI

3.2. ARIADNE

ARIADNE (*Alliance of Remote Instructional Authoring & Distribution Networks for Europe*) es el repositorio de un proyecto de la Unión Europea que tiene como objetivo fomentar el intercambio de experiencias en el área de la educación abierta y a distancia.

Consiste en una red europea de recursos educativos distribuidos alrededor del cual se han creado una serie de herramientas que ayudan a la compartir y reutilizar el material educativo. La Figura 10 muestra el acceso a la federación ARIADNE.

Figura 10 – Interfaz de consulta de ARIADNE

Provee una plataforma para compartir la información a través de la instalación de un servidor o nodo, mediante el cual se pueden almacenar y consultar los objetos de aprendizaje en línea que se generan en la comunidad, teniéndose así una base de OA distribuida en todos los nodos que componen la plataforma.

Un nodo de ARIADNE está formado por un repositorio local de almacenamiento (KLP) y un servidor de cursos (AMI/ALI), pudiendo estar ambos en la misma PC o distribuido en distintas computadoras.

El repositorio local de conocimiento se almacena todos los objetos de aprendizaje y sus metadatos, mientras que el servidor de cursos provee una interfaz Web para la administración y consulta sobre el repositorio.

Todos los nodos locales se comunican con un nodo especial, el nodo central, encargado de mantener la información actualizada de todos los objetos de aprendizaje y así poder compartir la información entre nodos locales. Para esto una vez al día el nodo central solicita a los nodos locales todos los OA dados de alta a partir de la última consulta y actualiza su catálogo de OA, este proceso es conocido como replicación. La Figura 11 ilustra lo antes mencionado:

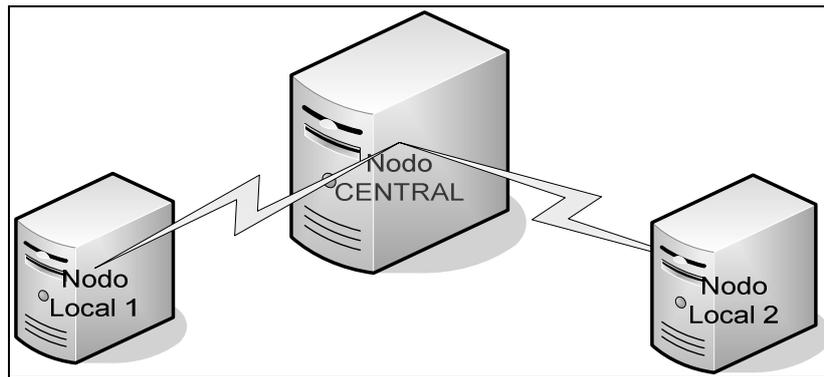


Figura 11 – Distribución de ARIADNE

Cada nodo local presenta una estructura bien definida, donde las componentes más destacadas de la arquitectura son:

- Gestor de KPS: se encuentra junto a la capa de datos almacenando los OA y los metadatos que los describen.
- SILO (Search and Index Objects): brinda el acceso al KPS.
- WEBLE (Web-Based Learning Environment): brinda herramientas para gestión de los cursos y acceso al KPS.
- KPS Client: brinda el acceso al KPS.

La Figura 12, muestra la distribución de los mismos en las diferentes capas que componen la arquitectura. También se pueden ver las dependencias entre los componentes y las dependencias entre los componentes y software.

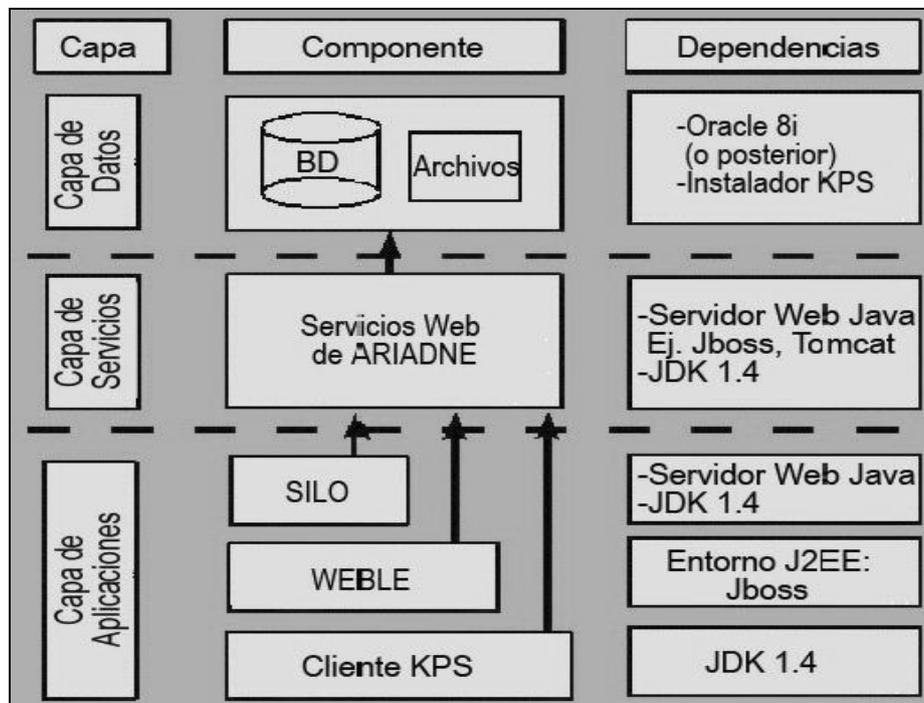


Figura 12 – Componentes de la Arquitectura de ARIADNE [17]

Otra de las características de ARIADNE es la búsqueda federada, es decir, permite realizar búsquedas en repositorios externos donde los lenguajes de consulta de cada repositorio no tienen porque ser los mismos, como por ejemplo el repositorio MERLOT (*Multimedia Educational Resource for Learning and Online Teaching*). Estas búsquedas se realizan utilizando SQI de manera transparente al usuario y resolviendo la heterogeneidad con los repositorios que desean unirse a la federación. Se puede ver cómo se lleva a cabo esta búsqueda, así como de qué forma se accede a cada repositorio de la federación:

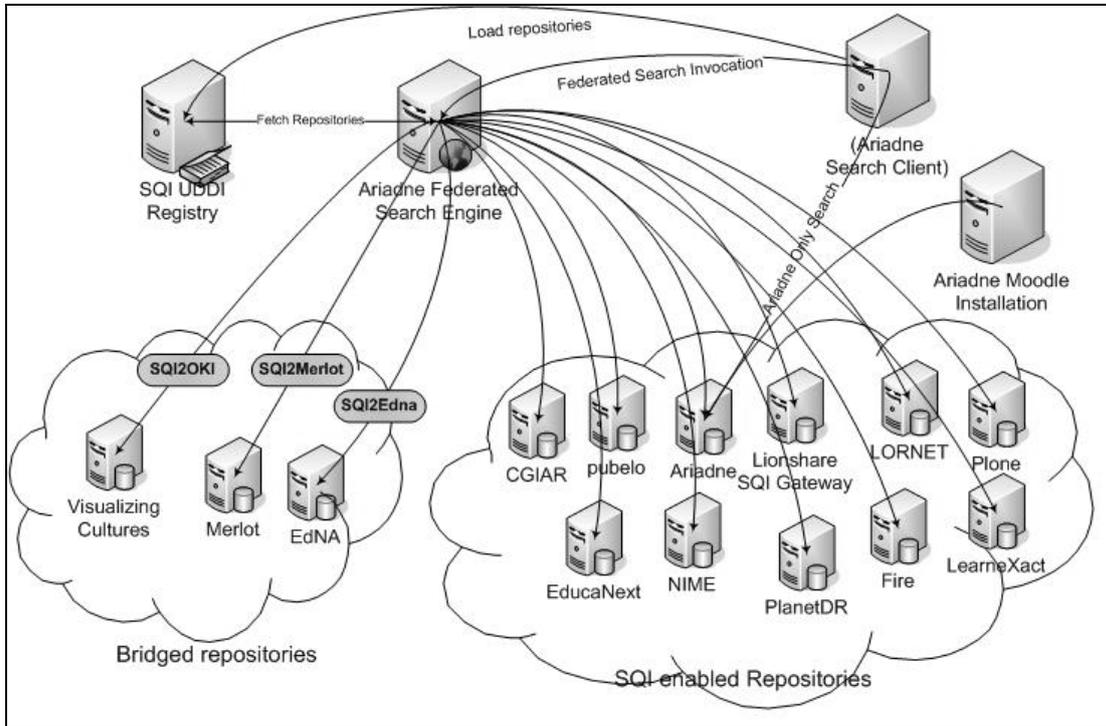


Figura 13 – Búsqueda Federada de Ariadne [26]

Como se puede ver la búsqueda federada se lleva a cabo a través de la API de consulta SQI, donde cada repositorio expone a través de servicios Web la API de consulta SQI. Por medio de esta se realiza la comunicación entre Ariadne y los repositorios.

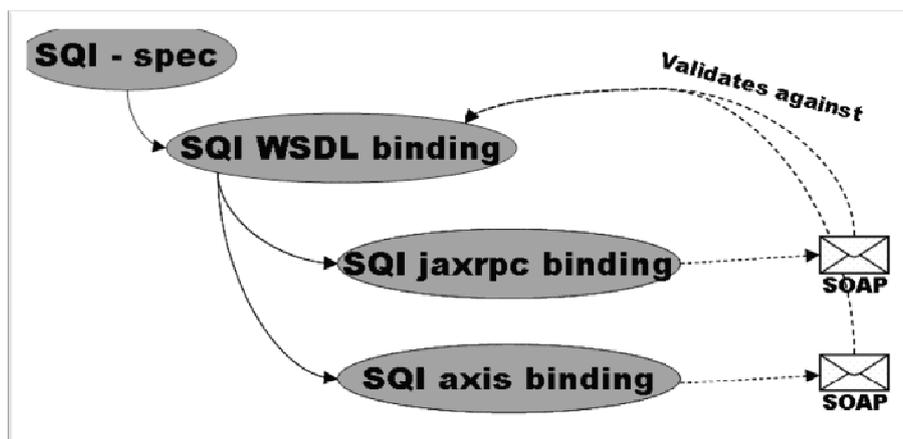


Figura 14 – API SQI a través de servicios Web [26]

Además de brindar la funcionalidad de descarga de contenido ofrece la posibilidad de exportar sus metadatos en LOM, logrando así una mayor reutilización de los OA con el

único inconveniente que el usuario final tenga que realizar una transformación de LOM al tipo de metadatos que use su repositorio.

En conclusión, uno de los puntos a destacar de esta arquitectura son las búsquedas federadas sobre repositorios heterogéneos en lenguaje de consultas a través de la API de SQI, de esta manera un repositorio para pasar a ser parte de la federación debe de exponer una interfaz SQI a través de servicios Web y devolver los metadatos de los OA en la especificación LOM. También brinda, una vez realizada la búsqueda, el filtrado de los metadatos obtenidos de acuerdo a los valores ya fijados para los metadatos de las categorías de LOM (Catalogo, Tipo, Formato, Contexto, Lenguaje).

La principal desventaja es que impone una arquitectura y plataforma fijas para los repositorios que se quieren integrar a la federación, brindando poca flexibilidad al uso de otro tipo de tecnología. Otra desventaja, no resuelve la heterogeneidad de los metadatos, exigiendo a cada repositorio que se quiera integrar a la federación que sus metadatos deben estar especificados siguiendo la representación LOM. En este punto cabe destacar que existe una marcada tendencia en el área de utilizar el esquema LOM para representar los metadatos de los objetos de aprendizaje.

Por otra parte la interfaz Web de consulta es muy poco intuitiva y carece de sencillez de uso para el usuario final. Esto ocasiona varios inconvenientes, como por ej. puede ocasionar que el usuario no haga un debido uso de las posibilidades que brinda la federación o deje de utilizarla.

3.3. GLOBE

GLOBE (*Global Learning Objects Brokened Exchange*) es un consorcio internacional integrado por EEUU, Japón, Europa, Canadá y Australia [32]. Tiene como objetivo el compartir recursos a través de una red distribuida de objetos de aprendizaje estandarizados.

Los repositorios que forman parte de GLOBE son ARIADNE, EdNA online (*Education Network Antralia*), NIME (*Nacional Institute of Multimedia Education*), MERLOT y LORNET.

La interoperabilidad con los repositorios que forman parte de esta federación, es resuelta, al igual que en ARIADNE, a través de la API SQI. El esquema de metadatos usado por esta federación, sigue el estándar LOM.

Los parámetros de consulta que se pueden especificar en un búsqueda federada son: palabra clava a buscar y los repositorios en los cuales se desea llevar a cabo la búsqueda. La Figura 15 muestra como lo antes mencionado.

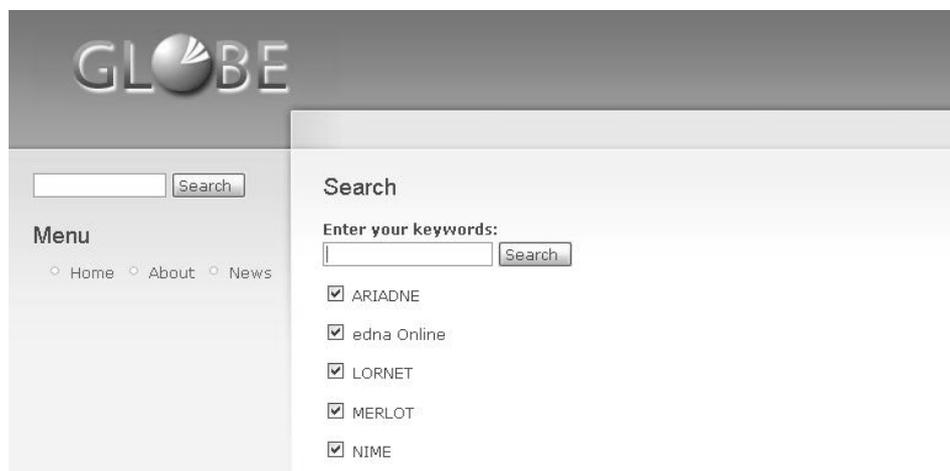


Figura 15 – Interfaz de consulta GLOBE

Una de las principales ventajas con respecto a las federaciones vistas hasta el momento, es que no impone una arquitectura de los repositorios que forman la federación, así como tampoco de la plataforma sobre la que este implementado el repositorio. En estos dos aspectos brinda la flexibilidad en la integración de un nuevo repositorio a la federación, de forma que el nuevo repositorio deberá brindar acceso a través de la API de consulta SQI y sus metadatos deben estar en el estándar LOM; pero nada exige en cuanto a la plataforma sobre la que este implementado el repositorio.

Una desventaja de la federación, es que no permite especificar una búsqueda por valores en los metadatos que describen los OA, no logrando tener una búsqueda demasiado específica y por lo tanto influyendo en la calidad de los resultados devueltos de la consulta. En este sentido no se hace un real aprovechamiento de los metadatos de un objeto de aprendizaje, dificultando la tarea de localización del material deseado por parte del usuario final. Otra de las desventajas, es que no ofrece acceso a todos los metadatos del objeto de aprendizaje, sino que únicamente permite acceder al contenido asociado al OA.

3.4. AGREGA

La federación consiste en un conjunto nodos interoperables, cada uno de estos nodos se corresponde a un repositorio de objetos de aprendizaje los cuales tienen una arquitectura particular tal como se detalla a continuación.

En cada uno de los nodos se puede buscar, visualizar y descargar contenidos. En el caso de la búsqueda se puede hacer una búsqueda en el nodo local o una búsqueda global, es decir en todos los nodos que integran la federación. A su vez es posible filtrar las búsquedas por: propiedades de contenido, área curricular, por tesaurus y por ámbito.

El acceso a la federación puede realizarse de dos formas, como usuario no registrado o como usuario registrado. En caso de hacerlo como usuario no registrado se podrá buscar, visualizar y descargar de lo contrario se tiene acceso a todas las funcionalidades de la herramienta, pudiendo agregar nuevos contenidos y proponerlos para compartir.

El acceso a la federación es independiente del navegador o dispositivo usado para acceder.

La arquitectura de cada nodo Agrega se compone de un conjunto de módulos especializados interoperando unos con otros. Sigue la filosofía Service Oriented Architecture (SOA) donde los dos grandes bloques de elementos lógicos, el nodo de objetos de aprendizaje y las aplicaciones clientes se integran usando como interfaz un conjunto de servicios. El modelo seguido se sustenta sobre tres pilares: interoperabilidad, especialización y evolución.

En lo que refiere a los dos bloques de elementos lógicos se tiene un bloque orientado al usuario que consta de [19]:

- El portal MEC, que permite a sus visitantes el acceso a los contenidos públicos albergados en cualquiera de los nodos de la red. A él accede cualquier usuario general, no registrado en ningún nodo autónomo.
- El portal para cada CCAA, desde el que se tendrá acceso a los datos públicos de los otros nodos, y permitirá poner a disposición de la comunidad educativa contenidos adicionales de cada CCAA. A él acceden usuarios autenticados en cada nodo autónomo.

El bloque orientado al nodo donde se destaca:

- “Interfaz de Interoperabilidad”. Se basa en el estándar de interoperabilidad de Repositorios de IMS (IMS-DRI), que proporciona la funcionalidad básica para

explotar los objetos digitales presentes en el repositorio (presentar/almacenar,y solicitar/entregar). Se ha implementado usando servicios Web, constituyendo los servicios de una arquitectura orientada a servicios (SOA).

- “Búsquedas de contenidos”. La búsqueda de contenidos entre repositorios se ha implementado usando la especificación “Simple Query Interface” (SQI). Respecto al mecanismo interno de búsqueda en un nodo, se basa en la indexación de un subconjunto de los metadatos los cuales son indexados a través de una herramienta denominada Lucene, que es la que se usa posteriormente para llevar a cabo las búsquedas.
- “Almacenamiento de contenidos”. Los contenidos almacenados se encuentran empaquetados conforme al estándar SCORM 2004, aunque se proporcionarán paquetes en los formatos, SCORM 1.2 e IMS-CP [4]. La información de catalogación se almacena conforme al estándar LOM v.1.0 en español (LOM-ES). En el interior de los nodos los objetos se encuentran desempaquetados dentro de un sistema jerárquico de ficheros.
- “Gestión de contenidos”. Para realizar la gestión de los objetos digitales se proporciona un conjunto de herramientas que facilitan la creación, catalogación, validación, publicación, visualización, etc. de los objetos digitales. La interfaz y funcionalidad de ésta se adapta atendiendo al perfil del usuario (docente, gestor, etc.) y al grado de conectividad disponible (off-line, on-line ligado a un nodo o independiente de los nodos).

La Figura 16 ilustra la arquitectura presentada:

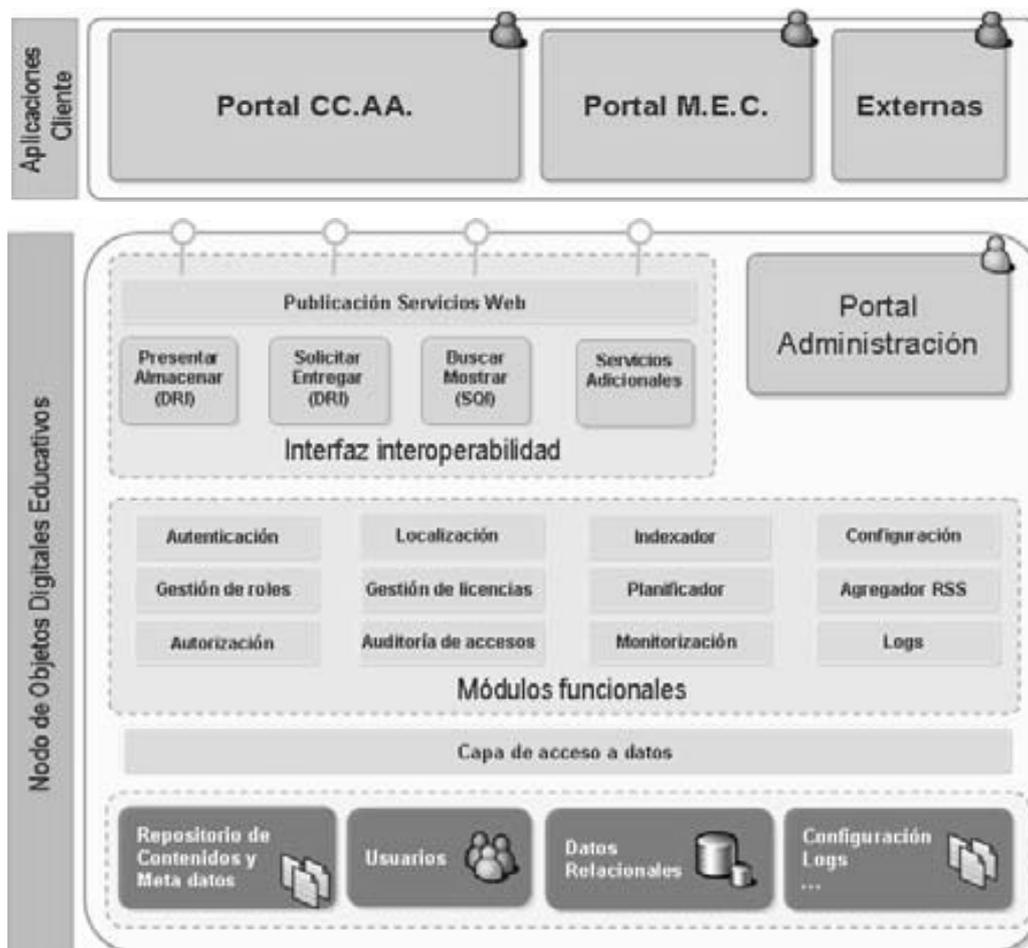


Figura 16 – Arquitectura de AGREGA [19]

Una de las ventajas del repositorio AGREGA, es que al igual que en el caso de Ariadne, comparte sus contenidos permitiendo el acceso a través del API SQI haciendo fácil de integrar a una federación. Otra, es el contenido rico en metadatos, tanto en calidad como en cantidad.

Se puede ver como una desventaja el no permitir integrar repositorios heterogéneos, ya que al momento de la integración el nodo debe cumplir con la estructura indicada en la arquitectura y con las tecnologías definidas a priori.

3.5. CORDRA

Se trata de “un sistema abierto, basado en estándares modelo para la forma de diseñar y aplicar sistemas de software con los fines de localización, intercambio y reutilización de contenidos de aprendizaje a través de la creación de federaciones de repositorios de contenidos de aprendizaje” [20].

Tiene como objetivo el compartir los objetos de aprendizaje que residen en los diferentes repositorios interoperables que federan el Sistema a través de un catálogo de metadatos que se detallará a continuación.

Los componentes de CORDRA son principalmente:

- *Local Content Repositories*: son los repositorios locales que contienen los objetos de aprendizaje y sus metadatos.

- *System Repositories*: forma parte de los sistemas de CORDRA. Mantiene tanto el registro de repositorios y sistemas como el catálogo maestro.
- *Identifier System*: provee la infraestructura para la identificación de los objetos de aprendizaje registrados.
- *Common Services Infrastructure*: servicios administrativos usados para la implementación del modelo, por ejemplo, autenticación, derechos de autor, reglas de procesamiento, etc.
- *Applications*: interfaces y aplicaciones usadas para el manejo de los objetos de aprendizaje por parte de usuarios finales (ej: búsquedas, registro).

El funcionamiento parte de que los repositorios que quieren formar parte de la federación, se inscriben en el registro de repositorios, donde figuraran los metadatos de sus objetos de aprendizaje y así permitirá la ubicación de los mismos. Los registros federados mantienen un catálogo con todos los metadatos de los objetos de aprendizaje que contienen, mediante el cuál se pueden realizar las búsquedas de los objetos de aprendizaje manteniendo un índice de los mismos para agilizar dichas búsquedas.

A diferencia con otras federaciones, en lugar de distribuir las búsquedas por los distintos repositorios, CORDRA centraliza los metadatos de los diferentes repositorios mediante un proceso de inscripción por cada repositorio.

Es un sistema sólo de localización de contenidos y no de recuperación, por lo que la recuperación va a estar dada por lo métodos que provea cada repositorio para la obtención del objeto de aprendizaje.

Mediante un conjunto definido de servicios se da acceso a toda la funcionalidad del Sistema y mediante una interfaz de usuario se da acceso a estos servicios a usuarios finales.

Es importante destacar que si bien el modelo CORDRA indica cuál es el diseño a seguir, en ningún momento hace referencia en cómo llevar a cabo este diseño. Por otra parte ADL se encuentra construyendo el *ADL-Registry*, el cual tiene como objetivo permitir la búsqueda y publicación de los objetos de aprendizaje.

Al no existir una implementación del modelo planteado surgen interrogantes como:

1. ¿Con que frecuencia se actualizará el catálogo de metadatos?
2. Dependiendo de las decisiones de actualización del catálogo, ¿cómo afectará esto al usuario en el acceso a objetos de aprendizaje agregados o modificados recientemente y que no figuran en el catálogo?
3. Al tratarse de un único catálogo conteniendo todos los metadatos de la federación donde su tamaño crece considerablemente, ¿cómo se verá afectada la eficiencia en las búsquedas? ¿en este punto no se tendrá un cuello de botella?
4. ¿Cuál es la estrategia a seguir para la integración de repositorios donde se cuenta con especificaciones de metadatos heterogéneas? ¿Los metadatos asociados a los OA serán entregados siguiendo la especificación del repositorio donde se encuentran o serán convertidos a una especificación fijada a priori?
5. ¿Cómo se mantiene consistentes metadatos que están replicados (dado que se encuentran en el catálogo y en el repositorio) y sufren modificaciones?

Se puede ver entonces que al no existir un prototipo de la solución sugerida surgen muchas interrogantes en puntos todavía a resolver, sin embargo una de las ventajas notorias de esta

solución es la centralización de la búsqueda resolviendo así la heterogeneidad en los lenguajes de consulta de los repositorios, de manera que cada repositorio sólo deberá proveer algún mecanismo solo de recuperación y no así de localización de contenidos. Pero como ya se vio esta misma centralización dependiendo de la solución implementada puede llegar a ser un cuello de botella en la búsqueda de contenidos a medida que la federación crezca.

3.6. Conclusiones

En el análisis llevado a cabo sobre las diferentes propuestas de federaciones que existen hoy en día, se puede ver que no hay una única solución, sino que diferentes propuestas cada una con las ventajas y desventajas mencionadas. También se puede observar una marcada tendencia a los estándares existentes en el área, como lo son LOM y SQI, para lograr sistemas extensibles, integrables e interoperables, así como la comunicación a través de servicios Web para lograr esto último. Además, es importante mencionar que el modo de acceso a cada una de las federaciones, tanto del usuario como también la forma que la federación accede a los repositorios, en todos los casos estudiados, es a través de Internet, debido a que se tratan de ambientes físicamente distribuidos.

A partir de lo analizado surge el cuadro comparativo (Tabla 5), el cual intenta mostrar las características y decisiones arquitectónicas de cada federación:

	Comunicación	Tipo de comunicación	Metadatos	Plataforma Heterogénea	Descarga OA	Empaquetador SCORM	Tipo de Búsqueda
<i>ARIADNE</i>	SQI	Servicios Web	LOM	No	Total	No	Avanzada
<i>GLOBE</i>	SQI	Servicios Web	LOM	Si	Parcial	No	Simple
<i>AGREGA</i>	SQI	Servicios Web	LOM-ES	No	Total	Si	Avanzada

Tabla 5 – cuadro comparativo de federaciones

El significado de cada columna del cuadro presentado, está dado por:

- *Comunicación:* refiere a como es resuelta la comunicación de la federación con los repositorios.
- *Tipo de comunicación:* muestra a través de qué medios se comunican la federación y repositorios
- *Metadatos:* esquema usado en cada repositorio para representar los metadatos
- *Plataforma Heterogénea:* refiere a la plataforma sobre la que están implementados cada uno de los repositorios. En el caso de tratarse de una plataforma heterogénea no hay ninguna condición exigida sobre la plataforma del repositorio para formar parte de la federación.
- *Descarga OA:* refiere a las posibilidades de descarga del OA, en el caso de ser total, permite acceder a la especificación de los metadatos y al contenido del OA. Para el caso parcial solo da acceso al contenido del OA.
- *Empaquetador SCORM:* refiere a si el repositorio brinda alguna posibilidad de empaquetamiento de los OA, generando paquetes SCORM.
- *Tipo de búsqueda:* en el caso de la búsqueda simple solo se permite buscar por palabra clave e indicar sobre que repositorios lanzar la búsqueda. La búsqueda avanzada

adicionalmente permite especificar valores para los distintos metadatos sobre los que se quiere buscar.

De acuerdo a lo analizado, se puede observar que la propuesta del Proyecto Agrega es una de las más completas en cuanto a las funcionalidades ofrecidas y que da solución a los problemas de integración e interoperabilidad de la federación con los distintos repositorios, considerando también las desventajas encontradas en esta solución.

4. Arquitectura Propuesta LOAD

4.1. Introducción

La arquitectura de software abarca decisiones importantes sobre:

- La organización del sistema software.
- Los elementos estructurales que compondrán el sistema y sus interfaces, junto con sus comportamientos, tal y como se especifican en las colaboraciones entre estos elementos.
- La composición de los elementos estructurales y del comportamiento en subsistemas progresivamente más grandes.
- El estilo de la arquitectura que guía esta organización: los elementos y sus interfaces, sus colaboraciones y su composición.

Sin embargo, la arquitectura de software está afectada no sólo por la estructura y el comportamiento, sino también por el uso, la funcionalidad, el rendimiento, la flexibilidad, la reutilización, la facilidad de comprensión, las restricciones de compromisos económicos y tecnológicos.

Se necesita una arquitectura de sistema para:

- Comprender el sistema
- Organizar el desarrollo
- Fomentar la reutilización
- Hacer evolucionar el sistema
- La utilización del sistema con un rendimiento óptimo

Cualquier sistema de un tamaño considerable evolucionará. Una vez esté en uso, el entorno cambiante provocará futuras evoluciones. Por lo tanto, el sistema debe ser fácil de modificar, esto quiere decir que los desarrolladores sean capaces de modificar partes del diseño o implementación sin tener que preocuparse por los efectos inesperados que puedan tener repercusión en el sistema.

En una primera parte se describe la arquitectura adoptando un enfoque funcional, analizando las funcionalidades básicas que deberá brindar, así como los actores implicados en la misma y sus relaciones, para finalmente dar un enfoque organizacional especificando el conjunto de capas en la que se ha estructurado la arquitectura y describiendo cada uno de los componentes de capa.

4.2. Funcionalidad soportada por la Arquitectura

En este punto se adoptará un enfoque funcional, para describir los requisitos funcionales que debe ofrecer este sistema de federación del LOAD, así como también los actores que hacen uso de estas funcionalidades. Después se pasará a detallar cada requisito en caso de uso, donde se explica con más detalle los pasos a seguir para completar con éxito la funcionalidad requerida. Por último se presenta el modelo de dominio del problema.

4.2.1. Especificación de requisitos

Una primera etapa para definir la arquitectura, es la especificación de requisitos. Mediante la especificación de estos requisitos se puede saber qué procesos son los que se llevan a

cabo en el dominio del problema y establecer los objetivos con los que debe cumplir la arquitectura.

A continuación se muestran los requisitos con los que tiene que cumplir la arquitectura:

Requisito	Descripción
R1	Realizar la búsqueda de metadatos
R2	Realizar búsqueda de metadatos avanzada
R3	Realizar la catalogación de los metadatos
R4	Proporcionar los metadatos solicitados
R5	Agregar un repositorio
R6	Modificar configuración de repositorio
R7	Baja de un repositorio
R8	Modificar configuración de búsqueda
R9	Alta de un usuario
R10	Modificar información de un usuario
R11	Dar de baja un usuario
R12	Realizar la autenticación de usuarios

Tabla 6 – Requisitos funcionales

4.2.2. Actores

Se describen a continuación los actores que utilizarán el Sistema.

Usuario Web: estos usuarios son los que acceden al sistema a través de una interfaz Web para realizar diferentes búsquedas sobre la federación.

Administrador: usuario encargado de todo lo relevante a la configuración del Sistema, como la gestión de otros usuarios administradores y la administración de los repositorios de la Federación.

Sistema: representa cualquier sistema que se comunica con la Federación para realizar diferentes búsquedas.

4.2.3. Casos de Uso

Se pasa a describir cada caso de uso correspondiente a los requisitos funcionales ya nombrados.

Para la descripción de cada caso de uso, se va a utilizar una plantilla cuyo formato se detalla a continuación:

Funcionalidad	<i>Nombre del caso de uso</i>
Objetivo	<i>Descripción informal de los objetivos</i>
Actores	<i>Actores que intervienen: principales y secundarios</i>
Precondiciones	Condiciones que deben cumplirse para llevarse a cabo
Pasos	<i>Secuencia de pasos para que se desarrolle con éxito. Se debe mostrar las interacciones entre los actores y acciones del Sistema</i>
Excepciones	<i>Excepciones para el caso de uso</i>

Tabla 7 – Plantilla de casos de uso

A partir de la plantilla anterior se pasa a documentar cada funcionalidad:

Funcionalidad	Realizar la búsqueda de metadatos
Objetivo	Buscar objetos de aprendizaje a través de los diferentes repositorios distribuidos.
Actores	Usuario Web y/o Sistema externo
Precondiciones	Indicar texto de los OA a buscar
Pasos	<ol style="list-style-type: none"> 1. Especificar texto a buscar 2. Seleccionar los repositorios sobre los que se quiere buscar 3. Realizar la búsqueda en todos los repositorios seleccionados 4. Presentar los resultados de la búsqueda al usuario
Excepciones	<p>Si se produce un error durante la búsqueda se le comunicara al usuario.</p> <p>Si no se encuentran OA para los parámetros especificados se le comunicara al usuario</p>

Funcionalidad	Realizar la búsqueda de metadatos avanzada
Objetivo	Buscar objetos de aprendizaje a través de los diferentes repositorios distribuidos
Actores	Usuario Web y/o Sistema externo
Precondiciones	Detallar la metainformación de los contenidos a buscar e indicar sobre que repositorios buscar
Pasos	<ol style="list-style-type: none"> 1. Completar el formulario con los campos de los metadatos a buscar 2. Seleccionar los repositorios sobre los que se quiere buscar 3. Realizar la búsqueda en los repositorios seleccionados 4. Catalogar los metadatos resultado de la búsqueda 5. Presentar los resultados de la búsqueda al usuario
Excepciones	<p>Si se produce un error durante la búsqueda se le comunicara al usuario.</p> <p>Si no se encuentran OA para los parámetros especificados se le comunicara al usuario</p>

Funcionalidad	Realizar la catalogación de metadatos
Objetivo	Clasificación de los metadatos, de forma de obtener un conjunto ordenado de metadatos según la coincidencias encontradas entre la metainformación detallada por el usuario y los metadatos resultado de la búsqueda.
Actores	Usuario Web y/o Sistema externo
Precondiciones	Que exista al menos un conjunto de metadatos resultado de una búsqueda
Pasos	<ol style="list-style-type: none"> 1. Obtener la cantidad de coincidencias para cada conjunto de metadatos devuelto. 2. Ordenar cada conjunto de metadatos de acuerdo a las coincidencias encontradas en cada conjunto de metadatos en orden descendiente.

Excepciones	Si se produce un error durante la catalogación se le comunicara al usuario.
--------------------	---

Funcionalidad	Proporcionar los metadatos solicitados
Objetivo	Poder acceder a los OA de los repositorios distribuidos
Actores	Usuario Web y/o Sistema externo
Precondiciones	Que exista al menos un conjunto de metadatos resultado de una búsqueda
Pasos	<ol style="list-style-type: none"> 1. Para cada conjunto de metadatos presentar al usuario los valores en los campos para los metadatos: <ul style="list-style-type: none"> • Título • Descripción • Ubicación • Repositorio donde se encuentra el OA 2. El usuario podrá seleccionar un conjunto de metadatos para poder ver los valores en los campos para cada uno de los metadatos (de ese conjunto).
Excepciones	Si el OA no puede ser accedido se le informará al usuario

Funcionalidad	Alta de un repositorio de OA
Objetivo	Agregar un repositorio a la Federación
Actores	Administrador
Precondiciones	El usuario está autenticado
Pasos	<ol style="list-style-type: none"> 1. Indicar el modo de acceso al repositorio 2. Indicar ubicación del repositorio 3. Indicar el modo de búsqueda, síncrono o asíncrono 4. Seleccionar lenguaje de consulta utilizado por el repositorio 5. Seleccionar esquema de metadatos devuelto en una búsqueda
Excepciones	En caso de no poder acceder al repositorio no se da el alta y se le comunica al usuario

Funcionalidad	Modificación de un repositorio de OA
Objetivo	Modificar configuración relevante de un repositorio
Actores	Administrador
Precondiciones	El usuario está autenticado
Pasos	<ol style="list-style-type: none"> 1. Modificar la ubicación del repositorio 2. Modificar el modo de autenticación 3. Modificar el lenguaje de consulta 4. Modificar el esquema de metadatos devuelto
Excepciones	En caso de modificarse alguno de los accesos a los servicios y no poder acceder, se le comunica al usuario.

Funcionalidad	Baja de un repositorio de OA
Objetivo	Dar de baja un repositorio de la Federación

Actores	Administrador
Precondiciones	El usuario está autenticado
Pasos	<ol style="list-style-type: none"> 1. Seleccionar el repositorio a dar de baja 2. El repositorio se da de baja en la Federación
Excepciones	Ninguna

Funcionalidad	Modificar configuración de búsqueda
Objetivo	Modificar los parámetros de búsqueda sobre los repositorios distribuidos
Actores	Administrador
Precondiciones	El usuario está autenticado
Pasos	<ol style="list-style-type: none"> 1. Modificar la cantidad de resultados devueltos en cada consulta 2. Modificar el tiempo de duración de cada consulta 3. Modificar criterios de ordenación en la catalogación
Excepciones	En caso de no poder efectuarse el cambio, se le comunica al usuario

Funcionalidad	Alta de un usuario administrador
Objetivo	Dar de alta un usuario administrador al Sistema
Actores	Administrador
Precondiciones	El usuario está autenticado
Pasos	<ol style="list-style-type: none"> 1. Se ingresa el nombre de usuario 2. Se ingresa la contraseña para el usuario
Excepciones	Si el nombre de usuario ya exista, se le comunica al usuario

Funcionalidad	Modificación de un usuario
Objetivo	Modificar la información de un usuario
Actores	Administrador
Precondiciones	El usuario está autenticado
Pasos	<ol style="list-style-type: none"> 1. Selecciona la información a cambiar
Excepciones	Ninguna

Funcionalidad	Baja de un usuario
Objetivo	Dar de baja un usuario
Actores	Administrador
Precondiciones	El usuario está autenticado
Pasos	<ol style="list-style-type: none"> 1. Selecciona el usuario a dar de baja
Excepciones	Ninguna

Funcionalidad	Realizar la autenticación de usuarios
Objetivo	Loguearse en el Sistema

Actores	Administrador
Precondiciones	Ninguna
Pasos	1. Ingresa el nombre de usuario 2. Ingresa la contraseña
Excepciones	En caso que el nombre de usuario o contraseña sean incorrectos, se le comunica al usuario

4.2.4. Modelo de Dominio

El modelo de dominio muestra los conceptos más importantes del Sistema expresados en forma de clases del dominio. En este problema a resolver consiste en la búsqueda de OA sobre repositorios heterogéneos y distribuidos en la red.

Como muestra la Figura 17, el Sistema se compone de un conjunto de repositorios y un conjunto de usuarios del Sistema. A su vez cada repositorio se compone de Objetos de Aprendizaje.

Las principal operación que se puede hacer sobre el Sistema, son las de realizar una búsqueda, la cual desencadena una búsqueda federada sobre los diferentes repositorios.

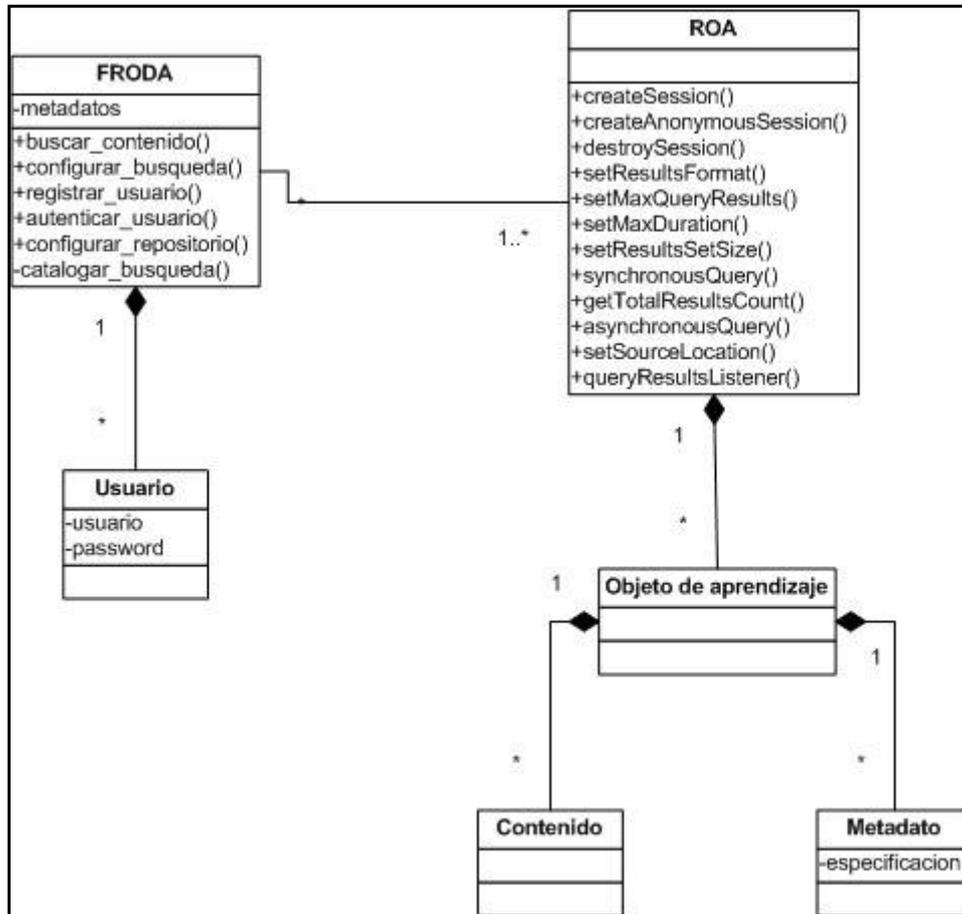


Figura 17 – Modelo de Dominio del problema

4.3. Organización en capas de la Arquitectura propuesta

En este punto se presenta la arquitectura como solución a los problemas planteados en los puntos anteriores. Para su construcción se siguió como patrón de diseño “Layers”, obteniendo una estructuración multicapas, donde cada capa sólo puede hacer referencia a componentes en capas inmediatamente inferiores. La arquitectura está basada en servicios Web y SOA (ver Sección 2.4).

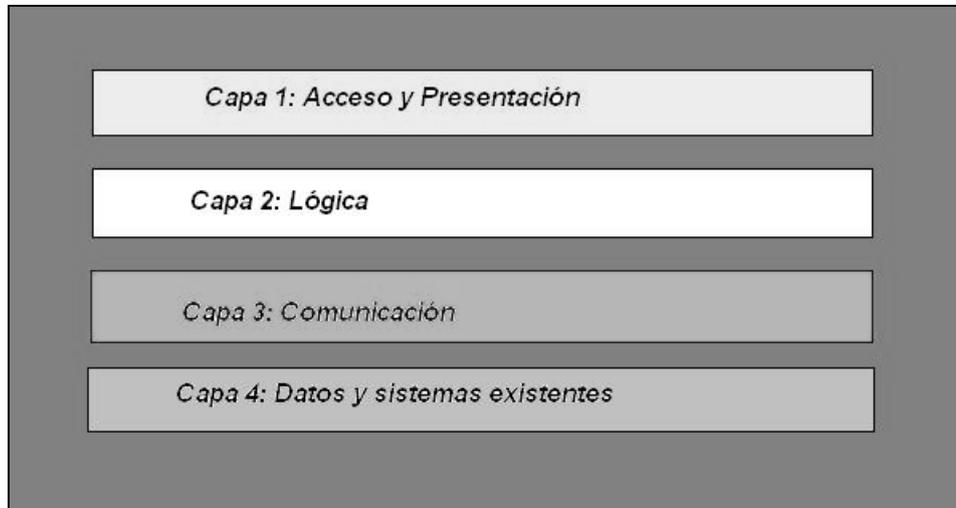


Figura 18 – Estructura en Capas de la Arquitectura

La arquitectura se descompone en las siguientes capas:

- ✓ *Capa de Acceso y Presentación:* esta capa es el punto de acceso para que usuarios puedan interactuar con el sistema y para que también otros sistemas externos puedan interactuar con la aplicación, de forma de poder llevar a cabo las funcionalidades propuestas.
- ✓ *Capa Lógica:* es esta capa se encierra la lógica de la aplicación, siendo responsable de contener la forma en que es llevada a cabo cada una de las funcionalidades.
- ✓ *Capa de Comunicación:* se encarga de gestionar la comunicación del Sistema con todos los sistemas externos (los repositorios de OA) a los que se quiere acceder, esta comunicación se realiza de manera bidireccional.
- ✓ *Capa de Datos y Sistemas existentes:* en esta capa se encuentra el modelo de datos de la aplicación, manteniendo toda la información relevante al Sistema como ser usuarios, configuraciones del Sistema e información de todos los sistemas externos a los que se quiere acceder, es decir los repositorios de OA.

A continuación se muestra en detalle cada una de las capas, se describen los componentes de cada una y de qué manera interactúan para llevar a cabo cada una de las funcionalidades ofrecidas por la aplicación. Para una mejor comprensión de la estructura de capas, se comienza la descripción partiendo de la capa de más bajo nivel y pasando a las que están en niveles superiores.

4.3.1. Capa de Datos y Sistemas existentes

El objetivo de la arquitectura es hacer transparente el acceso a los múltiples repositorios que conforman la arquitectura, por lo que el nivel más bajo se compone de todos los repositorios que contienen los objetos.

Almacena toda la información referente a los usuarios del Sistema y toda la información referente a cada uno de los repositorios que forman parte de la federación, como ser el modo de acceso, ubicación, lenguaje de consultas utilizado, etc.

Los servicios que brinda son todos aquellos encargados de la persistencia y recuperación de la información almacenada en el Sistema. Algunos de estos servicios son:

- ABM de Repositorios
- ABM de usuarios

Esta capa se abstrae del manejador de base de datos utilizado (DBMS), de forma que la aplicación es independiente de cuál sea el manejador elegido en la construcción del prototipo y teniendo así un sistema adaptable a cualquier DBMS.

4.3.2. Capa de Comunicación

Esta capa es la que relaciona los repositorios distribuidos con nuestra aplicación, conteniendo los servicios que se encargarán de la búsqueda federada.

En este punto pasaremos a describir, por un lado, cual es el alcance en la propuesta presentada y por el otro, las posibles extensiones soportadas por el LOAD.

En la propuesta presentada se accede a cada repositorio a través de servicios Web, desconociendo los detalles de implementación del repositorio al que se está accediendo. Cada repositorio deberá exponer sus servicios a través de la API SQI (detallada en puntos anteriores). De esta manera el Sistema podrá interactuar de manera transparente e independientemente de la plataforma, lenguaje de consulta o lenguaje de implementación en la que esté basado el repositorio.

Cuando se realiza una búsqueda federada, se realiza una invocación a los servicios de búsqueda locales de los repositorios a los que se tiene acceso. La comunicación entre la aplicación y el repositorio externo se hace a través de servicios Web.

El alcance de la propuesta arquitectónica da soporte a la comunicación entre el LOAD y los repositorios a través de servicios Web haciendo uso de la API SQI, donde cada conjunto de metadatos es presentado en el esquema LOM. Dada que una de las características deseadas en una arquitectura es la extensibilidad, para hacer posible la comunicación con otros repositorios que no presentan acceso a través de la API SQI y/o a través de servicios Web, alcanza con la incorporación de un módulo en esta capa que se ajuste al acceso brindado por el repositorio al que se quiere incorporar al LOAD.

Por otro lado, esta capa brinda a la Capa Lógica un servicio encargado del registro de un nuevo repositorio en la federación. Este servicio será encargado de verificar el acceso al repositorio de acuerdo a las configuraciones indicadas, en caso de acceder al repositorio se llama al servicio de la Capa de Datos encargado de persistir las configuraciones asociadas al repositorio y el repositorio es dado de alta en el LOAD. En caso contrario no da de alta el Repositorio y se le informa a la Capa Lógica.

4.3.3. Capa Lógica

En esta capa se tienen las funcionalidades detalladas en los casos de uso, es decir los servicios que el usuario invoca a través de la capa de acceso y presentación. Estos servicios desencadenan llamadas a servicios que se encuentran en las capas inferiores. A continuación se muestra cuales son estos servicios y como se lleva a cabo la interacción entre cada capa.

Búsqueda de contenidos: desencadena las llamadas a servicios que están en la capa de comunicación para llevar a cabo la búsqueda sobre los repositorios seleccionados. En este caso no se lleva a cabo una catalogación de los resultados obtenidos.

Búsqueda de contenidos avanzada: a partir de un conjunto de valores para los metadatos seleccionados por el usuario y sobre los repositorios indicados, se hacen llamadas a servicios de la capa de comunicación, recibiendo de ésta conjuntos de metadatos. Una vez obtenido los resultados de la búsqueda federada, se invoca al servicio de Catalogación en esta misma capa.

Catalogación de metadatos: consiste en, para cada uno de los conjuntos de metadatos evaluar las coincidencias entre los valores de los metadatos del conjunto (resultado de la búsqueda) y los valores de los metadatos especificados por el usuario. De esta forma se tendrá para cada conjunto un valor que llamaremos “indicador de coincidencia”, éste será calculado a partir de la cantidad de coincidencias encontradas en un conjunto de metadatos para el subconjunto de valores de metadatos indicado por el usuario. Se considera que un metadato (que llamaremos A) es coincidente con el mismo metadato (que llamaremos B), sí el valor A está contenido parcialmente y/o totalmente en el valor de B. En resumen, un metadato del subconjunto de metadatos indicado por el usuario es coincidente con el mismo metadato del conjunto de metadatos devuelto en la búsqueda, si el valor de primero está contenido en el segundo.

El alcance de la propuesta en este punto trabaja con el esquema de metadatos bajo el estándar LOM (por razones ya justificadas en puntos anteriores), tanto al momento del usuario especificar los valores para cada uno de los metadatos como al obtener los resultados de la búsqueda federada, pero no se limita a este esquema. Dado que algunas de las características deseadas en la arquitectura es la modularidad, adaptabilidad y extensibilidad, contempla el caso en que los resultados devueltos no sigan el esquema de metadatos bajo el estándar LOM, permitiendo adaptar el LOAD para trabajar con algún otro esquema de metadatos.

Registro de repositorios: a partir de recoger los datos indicados por el usuario, que describen el repositorio, en la capa de presentación, se llama al servicio en la capa de comunicación encargado de registrar el repositorio en el Sistema.

4.3.4. Capa de Acceso y Presentación

En este nivel se describe como es la interacción de un cliente con el Sistema. Se describen la principales interfaces y la invocación a los servicios de la capa Lógica.

Una vez que el usuario accede, puede llevar a cabo tres acciones:

Búsqueda de metadatos: en este caso deberá ingresar el texto o palabra clave a buscar y seleccionar el/los repositorio/s donde buscar. El resultado será presentado al usuario indicando la información más relevante para cada metadato encontrado.

Búsqueda avanzada de metadatos: en este caso el usuario deberá completar el formulario seleccionando en este caso los metadatos de interés y para cada uno de ellos los valores a

buscar. Además opcionalmente podrá seleccionar los repositorios sobre los que se va a llevar a cabo la búsqueda

Autenticación como administrador: una vez indicado el nombre de usuario y contraseña se accede a la interfaz de administrador, pudiendo llevar a cabo todas las operaciones vinculadas con este usuario, como el ABM de repositorios de OA, el ABM de usuarios administradores y el seteo de configuraciones generales de la búsqueda (máximo de resultados devueltos, tamaño devuelto en una consulta y tiempo máximo de ejecución de la consulta).

5. Implementación del Prototipo

En esta sección se describe la implementación del sistema, mostrando las decisiones que se tomaron para la selección de las tecnologías usadas y justificando el porqué de la selección.

5.1. Tecnologías

El sistema es *full* Web y fue desarrollado utilizando la plataforma PHP en su versión 5.2.10.

Para los aspectos visuales se utilizaron hojas de Estilo (CSS) de forma de lograr una presentación uniforme y minimizar el impacto ante un eventual cambio en la presentación. Adicionalmente se utilizó Java Script y Ajax (*Asynchronous JavaScript and XML*) para los aspectos funcionales de la presentación, evitando recargar toda la interfaz en cada petición al servidor, disminuyendo el tráfico en cada petición cliente-servidor y por lo tanto aumentando la velocidad en la comunicación. Por otra parte brinda al usuario una interfaz sumamente responsiva dado que no pierde de vista la información original del documento, sino que únicamente se actualizan algunas secciones de éste según el proceso de navegación. Para la el desarrollo de JavaScript y Ajax se utilizó el framework jQuery 1.3.2. Este ofrece numerosas ventajas al trabajar con estas tecnologías, principalmente simplificando el desarrollo con Ajax y Java Script en las aplicaciones Web, y garantizando la independencia del navegador, de manera transparente al desarrollador.

Para encapsular el acceso a los datos y asegurar la portabilidad entre los distintos manejadores de base de datos se utilizó la librería ADOdB para PHP. Con esta librería es posible cambiar el manejador de base de datos y su ubicación sin necesidad de reescribir cada llamada a la base de datos en la aplicación, asegurando así la portabilidad.

En cuanto a la comunicación del sistema con los repositorios que componen la federación, la cual se lleva a cabo consumiendo servicios Web, se utilizó la librería NUSOAP.

Como servidor de aplicación se usó Apache http Server en su versión 2.2.11.

Como manejador de base de datos se utilizó MySQL, versión 5.0.

La selección de este conjunto de tecnologías se realizó en base:

- Conocimientos previos y experiencia en el uso de algunas de las mencionadas. Estos conocimientos disminuyeron los tiempos de aprender el buen uso de una nueva tecnología y así poder enfocarse en resolver los problemas puntuales, como por ejemplo la comunicación del federador con los repositorios a través de servicios Web o la catalogación de los metadatos.
- Herramientas gratis y con el potencial para resolver los problemas planteados.

6. Conclusiones y Trabajos Futuros

6.1. Conclusiones

Se estudio el estado del arte en lo referente a los trabajos involucrados en el área de investigación de esta tesis, como ser: repositorios de objetos de aprendizaje y sus componentes, estándares, y todo lo vinculado a técnicas y tecnologías que permiten hacer a dos o más sistemas interoperables.

Luego se investigaron los diferentes trabajos vinculados con el objetivo de esta tesis, a fin de comprender mejor sus propiedades y detectar sus carencias, pudiendo así tener una visión crítica hacia cada una de ellos. Esta visión será fundamental al plantear la arquitectura del sistema LOAD.

Se cumplió con el objetivo principal, el de construir una federación donde los repositorios que la integran son heterogéneos en sus plataformas (software y hardware) y se encuentran en ambientes distribuidos, logrando la integración e interoperabilidad de cada uno de ellos con la federación. Para llevar a cabo lo antes mencionado no es necesario conocer en lo más mínimo ninguna de las características del repositorio, solo basta con conocer la ubicación del servicio Web de SQI y el esquema de metadatos utilizado por el repositorio.

Se propone una arquitectura SOA, que presenta características de ser una arquitectura adaptable y extensible, y se implementa una parte de la misma. Para dar lugar a esta, se siguieron de manera estricta los estándares propuestos en el área. Es muy sencillo incorporar un nuevo repositorio a la federación, solamente basta con especificarle a ésta, la ubicación del servicio Web del repositorio que se quiere agregar.

La implementación muestra la factibilidad de la arquitectura, logrando un prototipo que permita cumplir con el principal objetivo de la tesis, el de localizar objetos de aprendizaje distribuidos a través de la integración con otros sistemas, en este caso los repositorios de objetos de aprendizaje. Adicionalmente el prototipo brinda otras funcionalidades que hacen rica a la solución, como lo es la catalogación de los metadatos asociados a los objetos de aprendizaje.

Los repositorios con los cuales se llevo a cabo la integración son: Agrega, Merlot y Lornet. En estos casos se realizaron diferentes tipos de consultas y se evaluaron las diferentes funcionalidades provistas por SQI, obteniendo los resultados en cada caso sin demostrar inconvenientes.

Por último, el trabajo fue muy enriquecedor en distintos aspectos. En primer lugar enfrentarse a un problema con una temática desconocida, y realizar una investigación exhaustiva para poder cumplir con el objetivo del trabajo, donde se toman constantemente decisiones que deben estar debidamente justificadas. Por otra parte, es sumamente interesante el contexto donde esta planteado este trabajo, considero que los *e-learning* están creciendo a un ritmo acelerado y por lo tanto los trabajos en las distintas áreas que lo componen, donde quedan muchas cosas nuevas por hacer y otras ya hechas, por mejorar.

6.2. Trabajos Futuros

En esta sección se describen un conjunto de posibles mejoras a realizar sobre el prototipo de manera de obtener una herramienta de mayor utilidad.

Brindar acceso SQI

Brindar la funcionalidad al sistema de ser accedido por sistemas externos (por ejemplo, otras federaciones). Para esto bastaría con implementar las operaciones de la API de

consulta SQL. Estas operaciones deben ser accedidas a través de servicios Web, para lograr la integración e interoperabilidad de los sistemas en cuestión.

En este punto se podría agregar una operación más, la cual tenga como entrada un conjunto de valores de metadatos y devuelva los objetos de aprendizaje que cumplen con los parámetros indicados de la consulta. Así como también se especifique los repositorios sobre los cuales se quiere realizar la búsqueda. Con esto se acota el resultado de la búsqueda, reduciendo el tráfico entre los sistemas y por lo tanto agilizando la comunicación, punto no menor por tratarse de sistemas distribuidos que se comunican a través de Internet.

Almacenamiento local de OA

La idea en este punto consiste en tener un repositorio de objetos de aprendizaje local a la federación. Con esto se podrían almacenar objetos de aprendizaje, ya sea a demanda del usuario o los que se obtienen resultados de cualquier consulta.

Adicionalmente las búsquedas también incluirían a este repositorio, brindando al usuario la posibilidad de acceder a estos objetos de aprendizaje antes de lanzar una búsqueda federada por el resto de los repositorios, de forma de obtener resultados en menor tiempo. También es posible que el repositorio del que se obtuvo el objeto de aprendizaje este inaccesible o el objeto haya sido removido del repositorio, en ese caso se pueda obtener del repositorio local.

De implementarse este adicional, se deberá considerar un chequeo de repetidos en los resultados presentados al usuario, de manera que evitar presentarle dos objetos idénticos.

Reportes estadísticos

Todas las búsquedas que se realizan en la federación son almacenadas en la base de datos, con información relevante, por ejemplo, sobre que repositorios se hizo la búsqueda, que palabras claves se buscaron, sobre que metadatos se especificaron valores de búsqueda, tiempo de la consulta y cantidad de resultados obtenidos.

Sería de utilidad la generación de reportes que presenten, a partir de la información almacenada, información como:

- Cuáles son los repositorios “favoritos” al momento de lanzar una búsqueda
- De cuales se obtienen mayor cantidad de resultados
- Que metadatos son los más consultados
- Que tiempos ocupan las consultas
- Que repositorios tienen mejores tiempos de respuesta

Modificar contenidos

Consiste en dar al usuario la posibilidad de añadir valores a los metadatos de un objeto de aprendizaje. Adicionalmente se podrá modificar la metainformación asociada al objeto. Esto da como resultado un nuevo objeto de aprendizaje, que podrá ser almacenado, en caso de existir, en el repositorio local a la federación.

Soporte a Dublin Core

La federación trabaja con repositorios donde sus metadatos están representados en el esquema LOM. La arquitectura propuesta es fácilmente adaptable al esquema Dublin Core, ya que este último es considerado un subconjunto del esquema LOM. A continuación, en la Figura 19, se describe el mapeo necesario para adaptar el esquema de metadatos Dublin Core al esquema LOM, señalando la correspondencia entre cada uno de sus elementos. En

la primera columna se tiene el metadato correspondiente a Dublin Core y en la segunda columna su metadato correspondiente en el esquema LOM.

DC.Identifier	1.1.2:General.Identifier.Entry
DC.Title	1.2:General.Title
DC.Language	1.3:General.Language
DC.Description	1.4:General.Description
DC.Subject	1.5:General.Keyword or 9:Classification with 9.1:Classification.Purpose equals "Discipline" or "Idea".
DC.Coverage	1.6:General.Coverage
DC.Type	5.2:Educational.LearningResourceType
DC.Date	2.3.3:LifeCycle.Contribute.Date when 2.3.1:LifeCycle.Contribute.Role has a value of "Publisher".
DC.Creator	2.3.2:LifeCycle.Contribute.Entity when 2.3.1:LifeCycle.Contribute.Role has a value of "Author".
DC.OtherContributor	2.3.2:LifeCycle.Contribute.Entity with the type of contribution specified in 2.3.1:LifeCycle.Contribute.Role.
DC.Publisher	2.3.2:LifeCycle.Contribute.Entity when 2.3.1:LifeCycle.Contribute.Role has a value of "Publisher".
DC.Format	4.1:Technical.Format
DC.Rights	6.3:Rights.Description
DC.Relation	7.2.2:Relation.Resource.Description
DC.Source	7.2:Relation.Resource when the value of 7.1:Relation.Kind is "IsBasedOn".

Figura 19 – Mapeo Dublin Core a LOM

Lenguaje de consulta SQI

Una de las mejoras a largo plazo, consiste en adaptar la consulta sobre los repositorios al lenguaje de consulta “PLQL nivel 1”, este lenguaje es más potente que el usado actualmente (VSQL) ya que tiene como ventaja poder especificar una consulta por valores de metadatos, y no únicamente por términos.

Esto acelera mucho los tiempos de las búsquedas, disminuyendo el tráfico entre los repositorios y la federación, ya que únicamente el repositorio va a devolver los metadatos en cuyos valores se correspondan con los indicados en la consulta, a diferencia de lo implementado en la actualidad, donde se traen todos los metadatos que tienen alguna coincidencia para luego hacer un posterior filtrado por los que se corresponden a los valores especificados en la consulta.

El plazo en el que se pueda incorporar esta mejora, va a depender en gran medida de los repositorios que integran la federación, dado que por el momento ninguno soporta el lenguaje antes mencionado (PLQL1).

7. Bibliografía y referencias

- [1] Los ROA como soporte para los entornos e-learning
http://www.biblioweb.dgsca.unam.mx/libros/repositorios/bibliotecas_digitales.htm
Última visita: 01/07/2009
- [2] IMS Global Learning Consortium (IMS GLC)
<http://www.imsglobal.org/>
Última visita: 01/07/2009
- [3] Mediadores e interoperabilidad en e-learning
<http://jungla.dit.upm.es/~saguirre/publications/virtualEduca2004.pdf>
Última visita: 01/07/2009
- [4] Andrea Rodríguez, Raquel Sosa, .Asistente informático para la generación de Objetos de Aprendizaje.
Proyecto de Grado, Instituto de Computación, Facultad de Ingeniería, UDELAR, 2006
- [5] LACLO
<http://www.learningobjects2006.espol.edu.ec>
Última visita: 01/07/2009
- [6] IEEE Learning Object Metadata
<http://ltsc.ieee.org/wg12>
Última visita: 01/07/2009
- [7] IMS Digital Repositories Interoperability
http://www.imsglobal.org/digitalrepositories/driv1p0/imsdri_infov1p0.html
Última visita: 01/07/2009
- [8] IEEE Learning Technology Standards Committee (LTSC)
<http://ieeeltsc.wordpress.com/>
Última visita: 01/07/2009
- [9] E-learning América Latina
http://www.elearningamericalatina.com/edicion/mayo2_2004/na_2.php
Última visita: 01/07/2009
- [10] Un recurso pedagógico para e-learning,
<http://www.uvalpovirtual.cl/archivos/simposio2005/YankoOssandon-ObjetosDeAprendizaje.pdf>
Última visita: 01/07/2009
- [11] Article: Interoperability for Searching Learning Object Repositories
<http://www.dlib.org/dlib/january08/ceri/01ceri.html>
Última visita: 01/07/2009
- [12] Proyecto Agrega
<http://www.proyectoagrega.es>
Última visita: 01/07/2009

[13] Federador REDOUAA

<http://www.laclo.org>

Última visita: 01/07/2009

[14] Ariadne

<http://www.ariadne-eu.org>

Última visita: 01/07/2009

[15] Uso del Web semántico para la interoperabilidad semántica de recursos educativos

http://jungla.dit.upm.es/~saguirre/publications/telecom2005_interoperabilidad.pdf

Última visita: 01/07/2009

[16] SQI, Simple Query Interface

<http://nm.wu-wien.ac.at/e-learning>

Última visita: 01/07/2009

[17] Integración de REDOUAA a la Federación Latinoamericana de ROA

<http://ariadne.cti.espol.edu.ec/xavier/papers/Munoz-LACLO2007.pdf>

Última visita: 01/07/2009

[18] Simple Query Interface for Interoperable Learning Repositories

<http://nm.wu-wien.ac.at/e-learning/interoperability/www2005-workshop-sqi-2005-04-14.pdf>

Última visita: 01/07/2009

[19] Agrega- Plataforma de Objetos Digitales Educativos

<http://spdece07.ehu.es/actas/Canabal.pdf>

Última visita: 01/07/2009

[20] CORDRA

<http://cordra.net/docs/info/whatscordra/v1p00/info-whatscordra-v1p00.php>

Última visita: 01/07/2009

[21] OKI. Architectural concepts. OKI Open Knowledge Initiative, URL

<http://okiproject.org/filemgmt-data/files/OkiArchitecturalConcepts.pdf>

Última visita: 01/07/2009

[22] LOM – Learning Object Metadata.

http://ltsc.ieee.org/wg12/files/LOM_1484_12_1_v1_Final_Draft.pdf

Última visita: 01/07/2009

[23] The Web Services Platform

<http://webservices.xml.com/pub/a/ws/2001/04/04/webservices/index.html>

Última visita: 01/07/2009

[24] Descripción de servicios Web XML

[http://msdn.microsoft.com/es-es/library/77axffs8\(VS.80\).aspx](http://msdn.microsoft.com/es-es/library/77axffs8(VS.80).aspx)

Última visita: 01/07/2009

[25] Web Services Architecture

<http://www.w3.org/TR/2004/NOTE-ws-arch-20040211/>

Última visita: 01/07/2009

[26] ARIADNE

<http://www.ariadne-eu.org>

Última visita: 01/07/2009

[27] Representación del esquema LOM

http://es.wikipedia.org/wiki/Archivo:LOM_base_schema.png

Última visita: 01/07/2009

[28] Dublín Core

<http://dublincore.org/>

Última visita: 01/07/2009

[29] SCORM (Sharable Content Object Reference Model)

<http://adlnet.gov/scorm/index.aspx/>

Última visita: 01/07/2009

[30] SCORM, Una visión introductoria

<http://www.aproa.cl/1116/article-68376.html>

Última visita: 01/07/2009

[31] ADL – Advanced Distributed Learning

<http://www.adlnet.gov/>

Última visita: 01/07/2009

[32] GLOBE

<http://www.globe-info.org>

Última visita: 01/07/2009

[33] MERLOT (Multimedia Educational Resource for Learning and Online Teaching)

<http://www.merlot.org/merlot/index.htm>

Última visita: 01/07/2009

[34] LORNET

<http://www.lornet.org/merlot/index.htm>

Última visita: 01/07/2009