
Universidad de las Ciencias Informáticas

Facultad 4



Desarrollo de un módulo de interoperabilidad en
RHODA.

Trabajo de diploma para optar por el título de Ingeniero en Ciencias
Informáticas.

Autor: Yasmany Santalla Pereda

Tutores: Ing. Roxana Cañizares González

Ing. Javier Martín Soler

Ciudad de La Habana, Junio 2012

Declaración de Autoría

Declaramos que somos los únicos autores del trabajo “**Desarrollo de un módulo de interoperabilidad en RHODA**” y autorizamos a la Facultad 4 de la Universidad de las Ciencias Informáticas a hacer uso del mismo en su beneficio.

Para que así conste firmamos la presente a los ____ días del mes de _____ del año _____.

Autor: Yasmany Santalla
Pereda

Tutor: Ing. Roxana Cañizares
González

Tutor: Ing. Javier Martin Soler

PENSAMIENTO

“El mundo camina hacia la era electrónica...Todo indica que esta ciencia se constituirá en algo así como una medida del desarrollo; quien la domine será un país de vanguardia. Vamos a volcar nuestros esfuerzos en este sentido con audacia revolucionaria”

Ernesto Che Guevara

AGRADECIMIENTO

En primer lugar quiero agradecer a los integrantes del proyecto RHODA (Anabel, Angel, los profesores Yandris, Yailen, Leonardo, Orlando) y mis compañeras Bleidys y Karen, por todo el apoyo que me han ofrecido durante todo este tiempo.

A la valiosa ayuda de mis tutores Roxana y Javier, a los que siempre recordaré, por guiarme en la realización de este trabajo.

A Yaismel por toda su ayuda y todo lo que me enseñó.

A mis amigos de andadas en el cursar de estos cinco años (Pablo, Danir, Héctor, Ramiro, Vilson, Adrian, Alberto, Víctor, Raúl Milena, Yusimit, Darlenis, Yadira, Arelis, Milene y Angel) a quienes siempre llevaré en el corazón y a mis compañeros de apto y brigadas por soportarme.

A mis amigos del barrio Manuel, Ramón, Manolo, Humberto, Raúl, Yosbany, El baro, Yusniel, Danisbel, Yoan, Kiko, Yosmara, Yuri, Analina, Amanda y Madaylis, con los que siempre podré contar.

A toda mi familia, abuelos, tíos, tías, primos, primas y generaciones siguientes, los cuales me han ayudado en estos cinco años y los que siempre llevaré en mi corazón, mi mayor y más grande tesoro.

Por último quiero agradecer a la vida, por concederme unos padres maravillosos Alfredo y Maricela y ser fruto de ese amor puro, al igual que mis dos hermanitos Yeny y Yosbany, los cuales son todo para mí, gracias por existir, para andar juntos en el de cursar por la vida.

DEDICATORIA

A mis padres y hermanos por su amor incondicional, a mi familia y a todas las personas que han confiado en mí.

RESUMEN

Los repositorios de objetos de aprendizaje son herramientas clave para el almacenamiento, administración, localización y recuperación de los objetos de aprendizaje. Es de vital importancia que éstos puedan intercambiar información con otros sistemas en un entorno e-learning, contribuyendo a socializar el conocimiento. En el Centro de Tecnologías para la Formación (FORTES) de la Universidad de las Ciencias Informáticas (UCI), se ha desarrollado un repositorio denominado RHODA, el cual posee una limitada comunicación con otros sistemas, imposibilitando el eficiente acceso y reutilización de los recursos que almacena, así como el intercambio de información con otros sistemas.

La presente investigación tiene como objetivo principal desarrollar un módulo que permita la interoperabilidad de RHODA con otros sistemas que gestionen contenidos descritos con metadatos en un entorno e-learning, para lograr el intercambio de información. Para ello se analizan los referentes teóricos actuales relacionados con la interoperabilidad en los repositorios que gestionan contenidos educativos. En la solución se plantea un modelo integral de interoperabilidad compuesto por: colección, recolección, federación y publicación, sustentado en una arquitectura modular con la utilización de los estándares internacionales SQL, OAI-PMH y SPI.

Palabras Claves

Estándares, federación, interoperabilidad, modelos, objetos de aprendizaje, OAI-PMH, publicación, repositorios de objetos de aprendizaje, reutilización, recolección, SQL, SPI.

ÍNDICE

Tabla de contenido

INTRODUCCIÓN.....	1
CAPÍTULO 1.....	6
1.1. Interoperabilidad	6
1.2. La interoperabilidad en los entornos e-learning.....	10
1.3. Aplicaciones de un entorno e-learning.....	11
1.4. Repositorios de Objetos de Aprendizaje.....	14
1.5. Especificaciones y estándares de un entorno e-learning.....	17
1.6. Principales estándares de interoperabilidad para la comunicación y transferencia de contenidos.....	19
1.7. Arquitecturas para repositorios.....	24
1.8. Comportamiento de la interoperabilidad en repositorios del mundo y Cuba	26
1.9. Servicios Web	31
1.10. Metodología de desarrollo de software, tecnologías y herramientas a usar	35
Conclusiones parciales.....	39
CAPÍTULO 2.....	41
Introducción.....	41
2.1. Modelo integral de interoperabilidad en un repositorio que gestione contenidos educativos ..	41
2.2. Artefactos de los flujo de trabajo de las fases de inicio y elaboración de la metodología RUP	42
2.3. Propuesta de infraestructura tecnológica	43
2.4. Estándares de codificación.....	44
2.5. Modelo de Implementación	46
2.6. Interoperabilidad entre herramientas de un entorno e-learning según la propuesta.....	51
2.7. Pruebas de Caja Negra.....	52
2.8. Diseños de caso de prueba.....	52

Conclusiones Parciales	59
CONCLUSIONES	61
RECOMENDACIONES	62
BIBLIOGRAFÍA	63

INTRODUCCIÓN

En la actualidad el desarrollo de aplicaciones y soluciones informáticas está marcado por la utilización de distintas tecnologías y arquitecturas incompatibles entre sí, lo que imposibilita compartir recursos entre sistemas. La condición, gracias a la cuál sistemas heterogéneos logran comunicarse entre sí, es la interoperabilidad, formalmente no existe una definición única para el término antes expuesto, en coincidencia con los criterios aportados por los autores (Fuente, 2010), (IEEE, 1990), (IDABC EIF, 2008), (IFWG, 2005), el autor de la presente investigación define interoperabilidad como: *“la habilidad, capacidad o estado a alcanzar que tienen los sistemas, componentes o entidades de aplicaciones para transferir e intercambiar información, entendiéndose su estructura, logrando de una manera uniforme, independiente y eficiente el intercambio a través de las Tecnologías de la Información y las Comunicaciones (TIC), y utilizando la información intercambiada para el beneficio mutuo”*.

El desarrollo de las Tecnologías de la Información y las Telecomunicaciones (TIC), ha contribuido a mejorar todas las esferas de la sociedad. En la esfera de la educación surge como una novedosa y avanzada forma de aprendizaje en línea, que permite la informatización de los procesos de formación, capacitación e integración de diferentes aplicaciones. Fruto de esta informatización surge el aprendizaje electrónico o e-learning, que es definido como *“conjunto de tecnologías, aplicaciones y servicios orientados a facilitar la enseñanza y el aprendizaje a través de Internet/Intranet, que facilitan el acceso a la información y la comunicación con otros participantes”* (Red TNet España, 2006)

Desde los inicios del aprendizaje electrónico, la interoperabilidad entre las aplicaciones ha sido un factor clave, lo que permite intercambiar información con diferentes funcionalidades y objetivos. Dentro de un entorno e-learning existen diversos tipos de aplicaciones, como el Sistema de Gestión del Aprendizaje (LMS) por sus siglas en inglés Learning Management System, las Bibliotecas Digitales (BD), los sistemas de Gestión Académica (SGA), las Herramienta de Autor (HA), los Repositorios Digitales (RD), entre otros. Estas aplicaciones pueden trabajar independiente, pero solo cuando entre ellas existe un flujo de información, en otras palabras cuando son interoperables entre sí, es que alcanzan su verdadero potencial formativo.

Un RD constituye una herramienta eficiente para la gestión de contenidos, este se define como *“el lugar donde se almacena y mantiene la información digital”* (López Guzmán, 2005). Un RD presenta varias clasificaciones que serán citadas en la presente investigación y donde se hará mayor énfasis en los Repositorios de Objetos de Aprendizaje (ROA), conocido como *“repositorio que almacena colecciones de Objetos de Aprendizaje (OA) y sus descripciones (metadatos) o solamente guarda la*

descripción de OA almacenados en otros sitios, para poder acceder a su ubicación” (Montilva, y otros, 2010). El contenido almacenado en un ROA es conocido como OA el cual puede ser definido como: “recurso digital con una granularidad apropiada y una marcada intención formativa, compuesto por uno o varios objetos de información, con un único objetivo, descrito con metadatos y con un comportamiento secuenciado que asegure el correcto enlace entre los elementos de su estructura didáctica y que puede ser reutilizado en entornos e-learning” (Rodríguez, 2010)

Para lograr la interoperabilidad de los ROA y las diferentes aplicaciones de un entorno e-learning es necesario que su arquitectura esté basada en estándares.

- En la catalogación de los OA destacan los estándares: Metadatos de Objetos de Aprendizaje (LOM) por siglas en inglés, del IEEE y Dublin Core (DC), elaborado por Dublin Core Metadata Initiative (DCMI) por siglas en inglés, aunque existen muchos otros que son utilizados en los repositorios estudiados y que serán tratados en el capítulo 1.
- Para el empaquetamiento de los OA se emplean: Modelo de Referencia de Objetos de Contenido Compatibles (SCORM) y el estándar para el Empaquetado de Contenidos (IMS CP) del Proyecto de Sistemas para la Gestión del Aprendizaje (IMS) por sus siglas en inglés.
- Para la recolección de metadatos resulta muy útil, el Protocolo para la Recolección de Metadatos de la Iniciativa de Archivos Abiertos (OAI-PMH) por sus siglas en inglés.
- Para las búsquedas federadas se utilizan los estándares Open Knowledge Initiative (OKI) y la Interfaz Simple de Consulta (SQI) por sus siglas en inglés respectivamente. Los OA están usualmente descritos con metadatos, por lo que su búsqueda será más eficiente si se tienen en cuenta los mismos, para ello se hace necesario utilizar lenguajes de consulta estos son, ProLearn Query Language (PLQL), Contextual Query Language) (CQL), Query Exchange Language (QEL), Very Simple Query Language (VSQL), entre otros.

Entre los estándares relacionados con la arquitectura de un sistema destacan el estándar para la Interoperabilidad de Repositorios Digitales (IMS-DRI) y Content Object Repository Discovery and Registration/Resolution Architecture (CORDRA) por sus siglas en inglés respectivamente.

Para el acceso, reutilización, almacenamiento y distribución de OA existen varias aplicaciones de renombre mundial, entre las que se puede citar AGREGA¹, SMETE² y CAREO³, entre otros y redes de

1 Disponible en <http://www.proyectoagrega.es/default/home.php>

2 Disponible en <http://www.smete.org/smete>

3 Disponible en <http://www.careo.org>

repositorios internacionales como LACLO⁴, ARIADNE⁵, AGREGA⁶, GLOBE⁷, entre otros.

En su gran mayoría estas redes están constituidas por un número considerable de repositorios, permitiendo que de esta forma se reutilice la información almacenada en cada uno de ellos, es la interoperabilidad la condición gracias a la cual estos repositorios pueden comunicarse.

En Cuba, los RD se encuentran principalmente en los centros de Educación Superior, los cuales fueron recopilados de la consulta a los sitios oficiales de las universidades y las encuestas realizadas a dichas instituciones, tal es el caso de la Biblioteca Virtual de la Educación Superior (BIVES)⁸, el Repositorio de Recursos Educativos de la Universidad Virtual de la Salud (RREUVS)⁹, el Repositorio de Tesis de Doctorado en Ciencias Biomédicas y de la Salud¹⁰, de la red telemática Infomed y la Biblioteca Virtual de las Ciencias en Cuba, aunque se pueden encontrar en algunos institutos como por ejemplo, el Repositorio Digital Instituto de geografía Tropical¹¹.

La Universidad de las Ciencias Informáticas a partir del año 2010 cuenta con un ROA denominado RHODA, producido por el Departamento de Producción de Herramientas Educativas del Centro Tecnologías para la Formación (FORTES), encargado de almacenar y gestionar colecciones de OA con vistas a apoyar el proceso docente-educativo de la universidad.

Actualmente, RHODA posee una limitada comunicación con otros sistemas, solo permite intercambiar información con la Herramienta de Autor CRODA y el Sistema de Gestión del Aprendizaje MOODLE, ambos personalizados en la UCI. Esta comunicación se establece a través de una incipiente implementación de algunas funcionalidades del estándar SQI, además no se logra un intercambio de información con otros repositorios como los antes mencionados, que contribuya a compartir entre personas de diferentes países y entre las instituciones cubanas, el conocimiento existente y proporcionar diversas concepciones de temáticas específicas. De forma general no permite el intercambio de información con otros sistemas, como repositorios de recursos educativos y bibliotecas digitales existentes en las universidades cubanas. Estas dificultades constituyen una barrera que impide un eficiente acceso y reutilización de los OA que se encuentran en RHODA, así como el intercambio de información con otros sistemas. Por lo antes expuesto se plantea el siguiente **problema**

4 Disponible en <http://www.laclo.org/>

5 Disponible en <http://www.ariadne-eu.org/>

6 Disponible en <http://www.proyectoagrega.es/default/home.php>

7 Disponible en <http://www.globe.gov/>

8 Disponible en <http://bives.mes.edu.cu/>

9 Disponible en <http://recursos.uvs.sld.cu/index.php>

10 Disponible en <http://tesis.repo.sld.cu>

11 Disponible en <http://Dspace.geotech.cu>

a resolver: ¿Cómo propiciar el intercambio de información de RHODA con otros sistemas que gestionen contenidos descritos con metadatos en un entorno e-learning?

Se define como **objeto de estudio** la interoperabilidad en los repositorios que gestionen contenidos descritos con metadatos.

Para solucionar el problema y transformar el objeto de investigación se propone como **objetivo general** desarrollar un módulo que permita la interoperabilidad de RHODA con otros sistemas que gestionen contenidos descritos con metadatos en un entorno e-learning, para lograr el intercambio de información.

Objetivos específicos:

1. Elaborar el marco teórico conceptual de la interoperabilidad en repositorios que gestionen contenidos educativos descritos con metadatos.
2. Desarrollar los artefactos principales del flujo de implementación y prueba del módulo de interoperabilidad de RHODA.
3. Validar la propuesta de implementación del módulo de interoperabilidad para RHODA.

Teniendo como **campo de acción** los estándares de interoperabilidad en los repositorios que gestionen contenidos descritos con metadatos.

Para cumplir el objetivo propuesto, se formulan las siguientes **preguntas científicas:**

1. ¿Cuáles son los fundamentos teóricos que han caracterizado la comunicación de los Repositorios de Objetos de Aprendizaje con otras aplicaciones de un entorno e-learning a nivel mundial y en Cuba?
2. ¿Cuáles son los estándares que se usan en los repositorios existentes en el mundo y en Cuba?
3. ¿Qué estándares usar para lograr la interoperabilidad en RHODA con otros sistemas que gestionen contenido en un entorno e-learning?

Para dar cumplimiento a los objetivos se plantean las siguientes **tareas de investigación:**

1. Análisis del estado del arte para generar el marco teórico conceptual.
2. Recopilación de información a través de entrevistas y encuestas de los estándares que se usan en los repositorios que gestionan contenidos educativos existentes en Cuba.
3. Generar los artefactos del flujo de implementación y prueba.

4. Implementación del módulo de interoperabilidad en el repositorio RHODA.
5. Validación de la propuesta de implementación a través de los métodos definidos en la investigación.

El método investigativo empleado en la presente investigación es el método teórico, dentro del cual se utilizó el **análisis histórico – lógico** para realizar un estudio de los ROA existentes en Cuba y el mundo, sus características, así como las tendencias actuales de interoperabilidad entre aplicaciones de un entorno e-learning y **el método analítico - sintético** que permitió identificar los conceptos fundamentales y más importantes de los ROA, así como el análisis de toda la documentación relacionada con los principales estándares y especificaciones de interoperabilidad relacionadas con el tema.

Este trabajo de diploma presenta la siguiente **estructura capitular**:

Capítulo 1.- Fundamentación teórica: contiene un estudio del estado del arte de las herramientas, tecnologías y metodologías utilizadas en la actualidad para brindar soluciones a este tipo de problema. Este análisis posibilita la recopilación de información que se utiliza para definir los estándares, las librerías o frameworks y metodología a usar para la implementación del módulo.

Capítulo 2.- Implementación y validación de la propuesta: se describen los estándares a usar para el desarrollo del módulo de interoperabilidad, los artefactos correspondientes y se explican las principales funcionalidades y clases más importantes identificadas, se detalla cómo se realizó la validación de la propuesta, a través de pruebas de caja negra.

CAPÍTULO 1

En este capítulo se abordan las definiciones de interoperabilidad, así como las principales características de los repositorios. También se aborda los estándares más utilizados para lograr la comunicación de los repositorios con otras aplicaciones de un entorno e-learning. Se fundamenta la selección de las herramientas, metodología y tecnologías a utilizar para realizar la implementación del módulo de interoperabilidad de la versión 2.2 de RHODA.

1.1. Interoperabilidad

Desde el surgimiento de la informática, la interoperabilidad ha atravesado por diferentes momentos divididos en fases o generaciones. (SHETH, 1999)

Durante la primera generación, hasta 1985 la interoperabilidad era principalmente departamental y estaba enmarcada en el seno de una compañía, tenían en común que los sistemas implicados eran simplemente algunas bases de datos y ordenadores, en un área local o conectada directamente.

La segunda generación se extendió de 1985 a 1995, estuvo marcada por el significativo impacto de Internet y el advenimiento de la era Web, este alcance se amplía a todas las empresas, organizaciones e incluso a nivel inter-empresarial conectando decenas de ordenadores y repositorios de datos.

Desde 1996 hasta la actualidad se está desarrollando la tercera generación que se caracteriza por alcanzar mejoras significativas en las tecnologías de la comunicación, así como en las infraestructuras globales de información y de distribución computacionales. Además la dimensión de la distribución de datos ha logrado un alcance muy amplio, desde un sistema único a la globalidad.

Conceptos

Si bien existen múltiples definiciones de interoperabilidad, una de las más citadas, y que define la interoperabilidad a un alto nivel conceptual, es la que propone la IEEE definida como: “la habilidad de dos o más sistemas o componentes para intercambiar información y utilizar la información intercambiada” (IEEE, 1990)

El Vocabulario de Información y Tecnología ISO/IEC 2382 define el término como: “la capacidad de comunicar, ejecutar programas, o transferir datos entre varias unidades funcionales de forma que el usuario no tenga la necesidad de conocer la características únicas de estas unidades”.

El proyecto europeo IDABC (Interoperable Delivery of European reGovernment Services to public Administrations, Businesses and Citizens) para la interoperabilidad y el establecimiento de servicios de

gobierno electrónico a nivel europeo, afirma que *“la interoperabilidad es la capacidad de los sistemas de tecnologías de la información y la comunicación (TIC) y de los procesos de negocio que éstas soportan, para intercambiar datos y para ser capaces de compartir información y conocimiento”* (IDABC EIF, 2004). Esta definición ha sido ampliada, como *“la habilidad de organizaciones y sistemas dispares y diversos para trabajar juntos eficientemente para beneficiarse mutuamente en fines comunes”* (IDABC EIF, 2008)

Otras definiciones de relevancia proceden de la administración y el gobierno electrónico. La Interoperability Technical Framework (EITF) del Gobierno de Australia la caracteriza como *“la capacidad de transferir y utilizar información de una manera uniforme y eficiente a través de múltiples organizaciones y sistemas de tecnologías de la información”* (IFWG, 2005)

Existen otras que definen minuciosamente la interoperabilidad como: *“habilidad, capacidad o estado a alcanzar, que hace referencia a la relación entre dos o más recursos o sistemas (dispositivos hardware y de comunicación o componentes de software) de tecnologías de la información y de la comunicación, que precisan trabajar conjuntamente de forma fácil e incluso automática en el seno de una organización de múltiples organizaciones. Asimismo, las definiciones coinciden en señalar dos capacidades con las que deben contar los sistemas implicados: comunicarse entre ellos (para poder transferir información) y entender la estructura de la información que se transfiere entre las entidades (para poder utilizarla)”* (Fuente, 2010)

Formalmente no existe una definición única para el término antes expuesto, en coincidencia con los criterios aportados por los anteriores autores, el autor de la presente investigación define interoperabilidad como: *“la habilidad, capacidad o estado a alcanzar que tienen los sistemas, componentes o entidades de aplicaciones para transferir e intercambiar información, entendiéndose su estructura, logrando de una manera uniforme, independiente y eficiente el intercambio a través de las Tecnologías de la Información y las Comunicaciones (TIC), y utilizando la información intercambiada para el beneficio mutuo”*.

Se puede entender de esta manera que la interoperabilidad se manifiesta por:

1. La capacidad o habilidad de los sistemas para trabajar entre sí en tiempo real o programado.
2. La capacidad de la información para ser intercambiados entre diferentes sistemas (portabilidad)
3. La información intercambiada será utilizada para beneficio mutuo de los sistemas participantes.

En sistemas que gestionen información digital, dígame Bibliotecas Digitales, Repositorios

Institucionales, ROA, entre otros, la interoperabilidad puede estar representada en niveles. A continuación el autor hace un resumen de estos niveles según varias bibliografías consultadas (Krsulovic, 2012), (Gómez, 2007), ver **Figura 1**:

1. **Interoperabilidad de Infraestructura:** utilización de un medio físico/lógico que sirva como un canal que permita realizar los procesos de intercambio de datos, información, documentos, metadatos y objetos digitales. Independiente de la plataforma, soportados por protocolos de intercambio y acceso comunes a redes de datos tales como ISO-OSI y TCP/IP.
2. **Interoperabilidad Sintáctica:** En general se refiere a la capacidad de los sistemas de información para leer datos procedentes de otros similares y obtener una representación que pueda ser compatible. Esto se logra mediante la utilización de formatos/modelos estandarizados de codificación y estructuración de documentos y metadatos. Esto incluye el uso de lenguajes y metalenguajes estructurados (XML), junto con modelos de metadatos estandarizados (Dublin Core, Marc21, LOM, entre otros), cuyos elementos representan una sintaxis común entre los diferentes sistemas y serán detallados en acápites posteriores.
3. **Interoperabilidad Estructural:** Corresponde con los modelos lógicos comunes y la capacidad de los sistemas de información para comunicarse e interactuar en ambientes heterogéneos (en grupos abiertos/cerrados, con control centralizado/distribuido). Esto incluye la definición y utilización de protocolos especializados como OAI-PMH, SRU, RSS, entre otros, alguno de los cuales serán explicados en acápites posteriores.
4. **Interoperabilidad Semántica:** Se puede entender como la capacidad o habilidad de los sistemas que gestionen información, para intercambiar información basándose en un significado común de los términos y expresiones contenidos en los metadatos y documentos, con el fin de asegurar la consistencia, representación y recuperación de los contenidos. Esto involucra el uso de esquemas semánticos que incluyen vocabularios controlados.

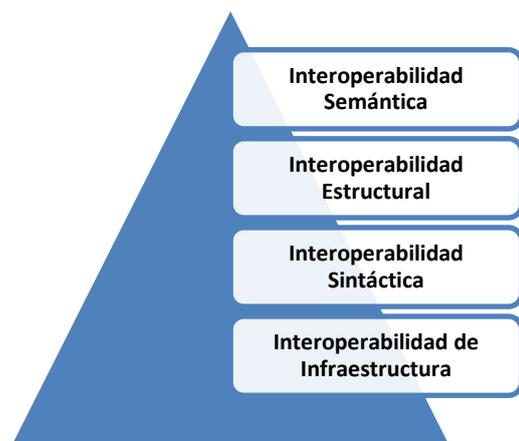


Figura 1: *Niveles de interoperabilidad*

Los investigadores de la National Science Digital Library (NSDL) estadounidense identificaron tres modelos de trabajo para la interoperabilidad de sistemas de información (Calzada Prada, 2010):

1. Colección (gathering). Es un enfoque de los motores de búsquedas, que permiten obtener un mínimo de interoperabilidad entre sistemas formalmente no cooperantes.
2. Recolección (harvesting). Basado en el acuerdo de un conjunto de sistemas para compartir determinados servicios.
3. Federación (federation). Permitiría obtener el mayor grado de interoperabilidad, pero requiere el acuerdo formal de un conjunto de sistemas para utilizar una serie de estándares y tecnologías comunes.

El autor coincide con lo antes expuesto, pero considera que además se debe incorporar un modelo en el cual se tenga en cuenta la publicación en otros sistemas de estos recursos, para de esta forma lograr un modelo global más general de la interoperabilidad en estos sistemas y que se pueda reutilizar esta información para beneficio mutuo, empleando para ello estándares. Permitiendo así que estén centralizados en un repositorio para su posterior acceso y reutilización ver **Figura 2**.



Figura 2: Modelo general de interoperabilidad en un repositorio

1.2. La interoperabilidad en los entornos e-learning

Con el advenimiento de la era digital, la modalidad de educación a distancia más acertada en la actualidad son los entornos de aprendizaje electrónicos, que emulan de forma eficiente los procesos de enseñanza-aprendizaje en la web. En la presente investigación al referirse al término aprendizaje electrónico o e-learning se usará la definición: *“conjunto de tecnologías, aplicaciones y servicios orientados a facilitar la enseñanza y el aprendizaje a través de Internet/Intranet, que facilitan el acceso a la información y la comunicación con otros participantes”* (Red TTnet España, 2006)

El creciente auge de estas tecnologías puede ser apreciado en el gran número de entornos educativos que existen a nivel mundial. Ellos son los encargados del acceso, reutilización, creación, almacenamiento y distribución de los contenidos educativos, en los cuales coexisten diferentes aplicaciones que interactúan entre sí y a la vez con otras plataformas educativas externas.

Un entorno e-learning está compuesto por varias herramientas, en la **Figura 3** se muestran algunas de las principales y cómo se relacionan entre sí.

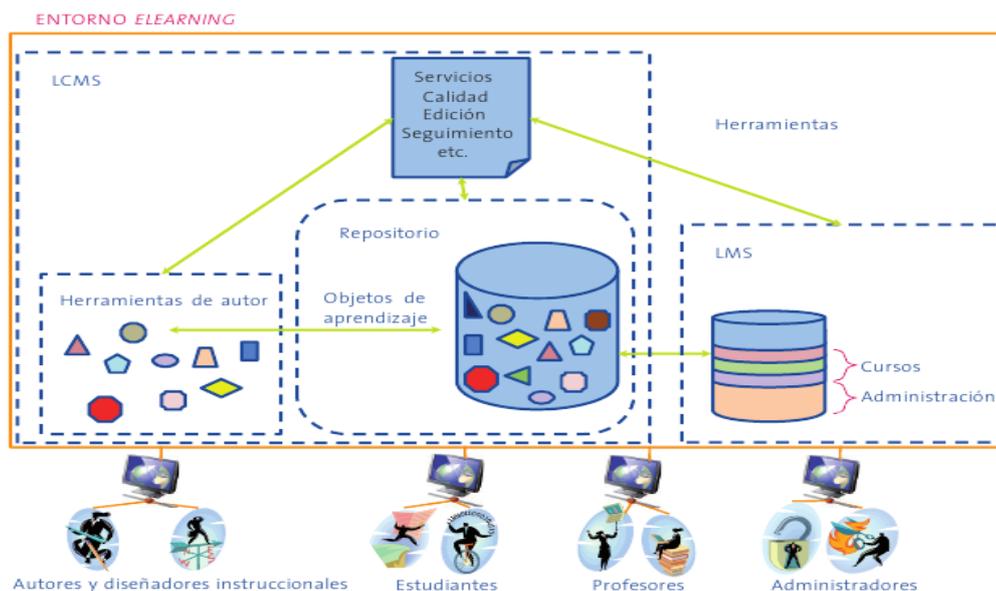


Figura 3: Integración de un LMS y un LCMS (tomado de Libro Blanco de la Universidad Digital, 2010)

Los entornos e-learning son complejos “ecosistemas” 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 labor compleja que requiere de esfuerzos importantes para lograr la comunicación y la transmisión de información, de forma parcial o global, entre aplicaciones y organizaciones (López Guzmán, 2005)

1.3. Aplicaciones de un entorno e-learning

En un entorno e-learning existen varias aplicaciones, dentro de las más utilizadas están:

Sistema de gestión de contenidos educativos (LMS)

El sistema de gestión del aprendizaje (LMS) por sus siglas en inglés, es el elemento central de la comunicación de un entorno e-learning. El mismo es un sistema basado en la web que permite el acceso a contenidos, la gestión de los recursos y la comunicación entre todos los actores implicados en el proceso (alumnos, profesores, administradores, entre otros).

Los LMS facilitan el aprendizaje distribuido y colaborativo a partir de actividades y contenidos pre elaborados, de forma síncrona o asíncrona, utilizando los servicios de comunicación de Internet como el correo, los foros, las videoconferencias y el chat. (López Guzmán, 2005)

Manejan todos los procesos que rodean el aprendizaje, pero resultan insuficientes a la hora de la creación, reutilización, administración o mantenimiento de los contenidos educativos, haciéndose necesaria la integración con un sistema especializado para la gestión de contenidos educativos LCMS por sus siglas en inglés. Aunque los LMS pueden almacenar estos contenidos la tendencia es a que exista un repositorio encargado de esta función, para ser accedidos y reutilizados por la plataforma LMS.

Sistema de gestión de contenidos educativos (LCMS)

Los sistemas de gestión de contenidos educativos o LCMS tienen su origen en los sistemas de gestión de contenidos (CMS) por sus siglas en inglés y cuyo objetivo es simplificar la creación y la administración de los contenidos en línea.

Los LCMS son sistemas multiusuario que están enfocados a aprobar, publicar, administrar y almacenar recursos educativos, a diferentes niveles, permitiendo de esta manera reestructurar la información y los objetivos de los contenidos de manera dinámica, para crear y modificar los objetos de aprendizaje que atiendan a necesidades y estilos de aprendizaje específicos.

Los LCMS se componen por una herramienta especializada en la creación de recursos educativos denominada herramientas de autor (HA) y un repositorio encargado del almacenamiento de los recursos educativos antes creados.

Herramientas de autor (HA)

El proceso de producción de un curso a distancia basado en el uso de las TIC requiere de conocimientos y habilidades que tienen su origen en el campo pedagógico e informático, por lo cual resulta bastante complejo la publicación de un curso, exigiendo del profesor cambios en su desempeño. Esta dificultad puede ser reducida con el empleo de software que automatizan parte del proceso, llamados HA. Las HA que crean OA al integrarse con repositorios aumentan su nivel funcional. La integración estará sustentada en el almacenamiento y reutilización de OA.

Como se puede apreciar la interoperabilidad entre los sistemas en un entorno e-learning es imprescindible para lograr un uso eficiente de las potencialidades de cada herramienta.

Repositorios Digitales (RD)

Los contenidos educativos, necesitan ser almacenados para su conservación y distribución, estos almacenes son los denominados repositorios digitales. El origen de este vocablo “*repositorio*” se deriva del latín “*repositorium*” que significa armario, alacena.

Este significado se generalizó en español y es definido por la Real Academia de la Lengua Española como: "*lugar donde se guarda algo*"; el término es aplicado al léxico específico de la informática para designar "*los depósitos de información digital*" (Real Academia Española)

Existen diferentes formas de clasificar los RD, una es en cuanto al origen y el tipo de información que almacenan, ejemplo son los Repositorios de Recursos Educativos (REA) y Repositorio de Objetos de Aprendizaje (ROA) (Chiarani, y otros, 2006), otras bibliografías consultadas la definen tres grupos principales: repositorios institucionales, repositorios temáticos y en cuanto a la información que almacenan.

Repositorio Institucional

En toda institución es vital almacenar la información creada para su posterior uso, razón por la cual surgen los repositorios institucionales. Teniendo como objetivos principales maximizar la visibilidad, el uso e impacto de la producción científica y académica de una institución y sus investigadores, facilitar y mantener el acceso a la información y como ésta impacta a la sociedad y preservar la información,

Un ejemplo de repositorios institucionales lo constituyen las bibliotecas digitales, estas nacen a partir de la figura de una biblioteca tradicional, con la misión de almacenar recursos digitales de información en un sitio web, de esta forma se van creando colecciones, con la intención de hacerlas disponibles para quienes se interesen por su consulta.

En la presente investigación el término biblioteca digital será definido como: "*cuando las colecciones de recursos digitales tienen un objetivo claro y se forman con una selección de contenidos organizados con un sistema descriptivo a través de metadatos (catalogación), y además se les asocian algunas facilidades para la búsquedas y uso de la información (servicios), estas colecciones se caracterizan como bibliotecas digitales*" (Borgman, 1999)

Existen algunas aplicaciones que generan bibliotecas digitales, las más importantes se mencionan a continuación:

- **DSpace**¹²: "*es un software de código abierto diseñado por el Massachusetts Institute of Technology (MIT) y los laboratorios de Hewlett-Packard (HP), para gestionar repositorios de ficheros, liberado en el 2002, como producto de una alianza de HP y el MIT. Es liberado bajo una licencia Berkeley Software Distribution (BSD) y totalmente compatible con el protocolo Open Archives Initiative-Protocol Metadata Harvesting (OAI-PMH). Hoy en día es muy utilizado*

¹² Disponible en <http://www.dspace.org/>

por diferentes instituciones en todo el mundo, como biblioteca digital” (Rodríguez Gairín, y otros, 2009)

- **Fedora**¹³: *“es un software de almacenamiento genérico, el sistema está diseñado de forma que sobre él se construyan repositorios y otras bibliotecas digitales que cumplen con el protocolo Open Archives Initiative (OAI). Lo han desarrollado la Universidad de Virginia y la Universidad de Cornell” (Rodríguez Gairín, y otros, 2009)*
- **Eprints**¹⁴: *“es una importante herramienta informática de software libre, desarrollada por un equipo dirigido por Christopher Gutteridge, de la Universidad de Southampton, que puede almacenar documentos en distintos formatos, es decir, imágenes, videos, audio, documentos en texto plano u otros” (Ramos, 2007)*

Repositorios Temáticos:

Los repositorios temáticos se encargan de almacenar los datos en cuanto a su temática, brindando una clasificación para diferentes ramas de la ciencia u otra disciplina, posibilitando una mayor organización de la información y además permitiendo el enriquecimiento de un área determinada del conocimiento, lo que contribuye al proceso de enseñanza-aprendizaje.

Repositorios de Recursos Educativos:

Los REA son depósitos de información digital de uso general, entre las herramientas más conocidas y con mayor aceptación que generan estos repositorios se encuentran: Dspace, Fedora, Eprints, entre otros, definidos como: *“los repositorios abiertos o de uso general se implementaron para favorecer la libre publicación de producción de documentos entre instituciones académicas. Para ello se crearon repositorios donde los artículos, tesis o documentos se almacenaban para evaluación de otros pares. El modo de revisión se ampliaba, de esta manera, de los comités científicos a todos los especialistas en el tema” (Chiarani, y otros, 2006)*

En el mundo existen ejemplos que ilustran claramente a estos repositorios, tal es el caso de Merlot, Temoa, Clarise, entre otros.

1.4. Repositorios de Objetos de Aprendizaje

Una distinción especial de los repositorios digitales son los ROA, considerados como almacenes de OA y extrapolados al contexto educativo se definen como: *“repositorio que almacena colecciones de*

¹³ Disponible en <http://fedora-commons.org/>

¹⁴ Disponible en <http://arxiv.org/>

Objetos de Aprendizaje (OA) y sus descripciones (metadatos) o solamente guarda la descripción de OA almacenados en otros sitios, para poder acceder a su ubicación” (Montilva, y otros, 2010)

El autor consultó varias definiciones citadas por López Guzmán C., 2005 en su tesis “*Los Repositorios de Objetos de Aprendizaje como soporte a un entorno e-learning*” la cuales transcribe a continuación:

- El programa CANARIE plantea que los ROA “*son un catálogo electrónico/digital que facilita las búsquedas en Internet de objetos digitales para el aprendizaje*” (CANARIE, 2001)
- El JORUM+ Project adopta la siguiente definición: “*Un ROA es una colección de OA que tienen información (metadatos) detallada que es accesible vía Internet. Además de alojar los OA los ROA pueden almacenar las ubicaciones de aquellos objetos almacenados en otros sitios, tanto en línea como en ubicaciones locales*” (JORUM+Project, 2004)

Las definiciones, en su sentido general no difieren mucho entre sí y dejan ver claramente que estos repositorios, sean bases de datos o catálogos, están creados para ser utilizados en un proceso de enseñanza, en la actual investigación al referirse al término antes expuesto utilizaremos la definición propuesta por (Montilva, y otros, 2010)

Tipos y características principales de los ROA

Por la forma en la que se concentran los OA, principalmente se identifican dos tipos de ROA: (Downes, 2004)(Riley, y otros, 2003)

- Los que contienen los objetos de aprendizaje y sus metadatos: En éstos los objetos y sus descriptores se encuentran dentro de un mismo sistema e incluso dentro de un mismo servidor **Figura 4a.**
- Los que contienen sólo los metadatos: En este caso el repositorio contiene sólo los descriptores y se accede al objeto a través de una referencia a su ubicación física que se encuentra en otro sistema o repositorio de objetos **Figura 4b.**

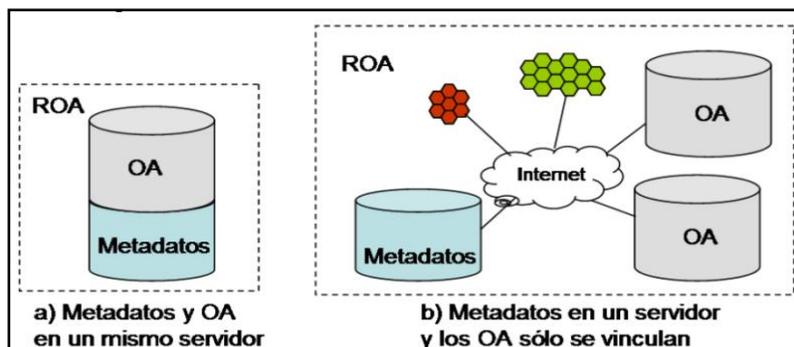


Figura 4: Tipos de ROA por la distribución de los OA y sus metadatos tomado de (López, 2005)

También es común encontrar repositorios mixtos, en los que se hace una combinación de estos dos tipos antes mencionados.

Los ROA más conocidos comúnmente funcionan de forma independiente, son aplicaciones con una interfaz web, un mecanismo de búsqueda y listados con algún tipo de clasificación. Otra clase de ROA operan sólo como módulos adicionales a otros productos (LMS o LCMS) que utilizan los contenidos de forma de forma exclusiva y sin que el usuario tenga acceso directo al repositorio.

Lo deseable es que los ROA tengan ambas capacidades, tanto ofrecer una interfaz web, para que los usuarios humanos puedan acceder a la colección, así como la capacidad de comunicarse directamente con las plataformas de aprendizaje y hacer posible la interoperabilidad entre sistemas de diferente naturaleza.

Por la forma en la que los catálogos de metadatos se organizan, se diferencian dos modelos de ROA (Downes, 2004): centralizados y distribuidos. Los más comunes son los centralizados (**Figura 5a**), en los cuales metadatos de los OA están contenidos en un mismo servidor, aunque el objeto esté localizado en alguno otro. El modelo distribuido (**Figura 5b**) opera a través de varios servidores, cada uno contiene diferentes grupos de metadatos y se comunican entre ellos para intercambiarlos.

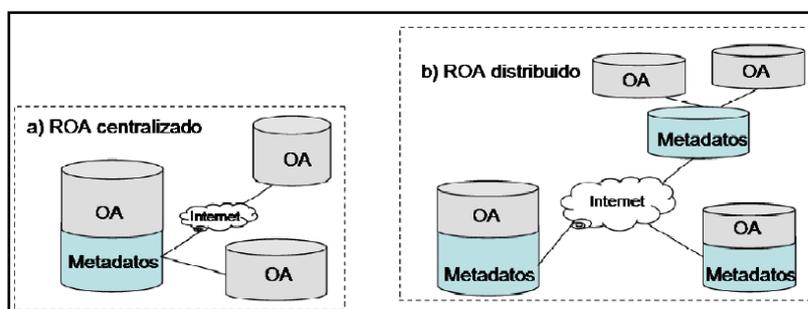


Figura 5: Tipos de ROA por la distribución de los metadatos tomado de (López, 2005)

Independientemente de las características que permiten clasificar a los ROA, lo más importante es que contribuyan a mejorar los procesos de gestión de OA dentro de los entornos educativos y que tengan “capacidades, tanto para ofrecer una interfaz web, para que los usuarios humanos puedan acceder a la colección, así como la capacidad de comunicarse directamente con las plataformas de aprendizaje y hacer posible la interoperabilidad entre sistemas de diferente naturaleza” (López Guzmán, 2005)

Las clasificaciones de los repositorios mencionadas anteriormente, no se adecuan a todos los entornos, presentando dificultades a la hora de determinar el tipo de información que almacenan, que en muchos casos es mixta. Por lo tanto, el autor de la presente investigación difiere en la clasificación propuesta por (Chiarani, y otros, 2006), y considera que los repositorios digitales se clasifican en Repositorios Institucionales, Temáticos y por el tipo de información que almacenan estos pueden ser REA, ROA entre otros la **Figura 6** ilustra con más detalle esta explicación.

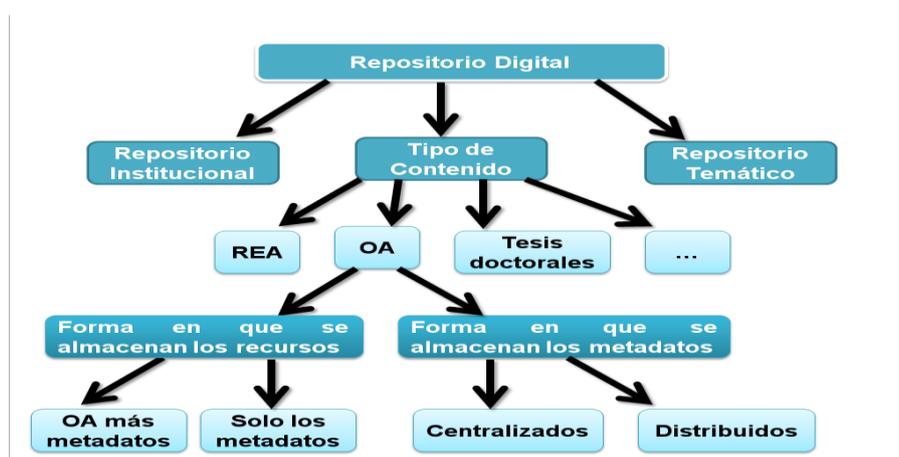


Figura 6: Clasificación de los repositorios digitales

Los repositorios digitales, almacenes de recursos digitales, por propia definición necesitan almacenar recursos que se encuentran en otras aplicaciones. Esta comunicación es factible, si los sistemas involucrados, emplean estándares y especificaciones comunes.

1.5. Especificaciones y estándares de un entorno e-learning

Para lograr la interoperabilidad de los ROA y las diferentes aplicaciones de un entorno e-learning es necesario que su arquitectura esté basada en estándares, a continuación se expondrán los principales estándares usados para describir y empaquetar recursos educativos, los cuales permitirán la reutilización de los recursos:

Estándares de catalogación

La necesidad de localizar, clasificar y reutilizar recursos digitales es la principal razón por la que distintas organizaciones se han involucrado en procesos de elaboración de esquemas de metadatos para establecer con estos un conjunto de reglas semánticas, sintácticas y de contenido que pretenden describir objetos de aprendizajes o recursos digitales. Dentro de estos esquemas se encuentra Dublin Core¹⁵, Canadian Core Learning Resource Metadata Application Profile (Cancore)¹⁶, LOM-ES¹⁷ y LOM¹⁸.

IEEE Standard for Learning Object Metadata (LOM)

El IEEE junto al LTSC, desarrolló en el año 2002 el estándar LOM (Learning Object Metadata) para la descripción interoperable de los OA. El propósito de este estándar multi-parte es facilitar la búsqueda, evaluación, adquisición y uso de los objetos educativos, por ejemplo, por alumnos, profesores o procesos automáticos de software.

Los esquemas para la descripción de recursos, proponen un conjunto de elementos de información, en el caso de LOM, posee sesenta y cuatro elementos de información, un número bastante grande que facilita una amplia descripción del OA. Algunos esquemas reúnen sus elementos relacionados dentro de diferentes grupos, a los que llaman categorías, para una mejor organización.

Los elementos de información de LOM se encuentran agrupados en nueve categorías estas son (General, Ciclo de Vida, Meta-Metadatos, Técnica, Uso Educativo, Derechos, Relación, Anotación, Clasificación) LOM se considera el más abarcador de todos los esquemas, ya que contiene una gran cantidad de metadatos. Su principal objetivo es la creación de descripciones estructuradas que permiten conocer el contenido del OA sin necesidad de abrirlo.

Dublin Core (DC)

Esquema de metadatos elaborado por la DCMI (Dublin Core Metadata Initiative), enfocado al ámbito bibliotecario, no es tan amplio como LOM en cuanto a posesión de elementos, puesto que solo posee quince y su utilización no es de carácter obligatorio, mientras que LOM cuenta con sesenta y cuatro.

Aun así Dublin Core sigue siendo uno de los estándares con mayor difusión a nivel mundial, por su simplicidad (tan sólo 15 elementos: Contributor, Coverage, Creator, Date, Description, Format,

15 Disponible en <http://dublincore.org>

16 Disponible en <http://www.cancore.ca>

17 Una adaptación de LOM creado y utilizado por la comunidad educativa de España.

18 Para consultar los 64 elementos propuestos por LOM www.uvs.sld.cu/archivos/lomv1spanish.rar/

Identifier, Language, Publisher, Relation, Rights, Subject, Source, Title y Type), cada elemento es opcional, puede repetirse y aparecer en cualquier orden. La principal desventaja de este esquema es que al ser aplicado para la descripción de un OA o de cualquier otro recurso educativo, resulta poco descriptivo, la información que brinda no es suficiente. Aunque es muy utilizado para describir objetos de información (OI) recursos que se almacenan en las bibliotecas.

Estándares de empaquetamiento

Instructional Management System Content Packaging (IMS-CP) por sus siglas en inglés es una variante de estándar para el empaquetamiento de contenidos promovida por IMS¹⁹. IMS-CP describe estructuras de datos que pueden ser usadas para intercambiar información entre sistemas que deseen importar, exportar, agregar, y desagregar paquetes de contenido (IMS, 2012)

SCORM, es la unión de un conjunto de normas dirigidas al empaquetamiento, distribución, reutilización y ejecución de contenidos educativos, enfocado a lograr una mayor interoperabilidad entre las diferentes herramientas de un entorno e-learning. (Baltazar, y otros, 2006). Esta norma cuenta con varias versiones, siendo la 1.2 y 2004 las más difundidas y utilizadas.

SCORM usa IMS-CP como modelo de agregación de contenido, como esquema de metadatos LOM u otro y la secuenciación de los contenidos está basada en los trabajos de la AICC²⁰ en esta área.

1.6. Principales estándares de interoperabilidad para la comunicación y transferencia de contenidos

La capacidad de comunicarse directamente con las plataformas de aprendizaje y hacer posible la interoperabilidad entre sistemas de diferente naturaleza, está dada por la utilización de estándares, especificaciones y protocolos, algunos de ellos son Simple Query Interface (SQI), Open Knowledge Initiative (OKI) para búsquedas federadas, Open Archives Initiative-Protocol Metadata Harvesting (OAI-PMH) y Open Archives Initiative-Object Reuse and Exchange (OAI-ORE) para la recolección e intercambio de metadatos, Simple Publishing Interface (SPI) para la publicación de OA, entre otros que serán explicados a continuación.

Simple Query Interfaces

¹⁹ IMS Global Learning Consortium es un grupo independiente, sin ánimo de lucro que inició su labor en 1997. Actualmente es el principal promotor de la creación de especificaciones abiertas dirigidas a la enseñanza electrónica. Su objetivo es que, a partir de estas especificaciones, se consiga la interoperabilidad de aplicaciones y servicios.

²⁰ AICC, Aviation Industry Computer Based Training Committee (Comité de la Industria de la Aviación para el aprendizaje basado en Computadoras), fue creado en 1992 y fue el primer organismo creado para desarrollar un conjunto de normas que permitiese el intercambio de cursos CBT (Computer Based-Training) entre diferentes sistemas.

Los ROA usan distintos modelos de datos para la gestión de los OA, por lo que para lograr la interoperabilidad se hace necesario el uso de un lenguaje común de consulta para la localización de OA, este lenguaje global debe ser traducido en cada instancia a un lenguaje de consulta local según la implementación que exista en el repositorio que la procesa, por lo que se hace necesario el uso de una interfaz de comunicación para ejecutar dicha consulta, en este diálogo media una fuente (instancia que realiza la consulta) y un destino (repositorio que procesa la consulta y devuelve un resultado) (VanNguen, 2007)

SQL es una interfaz para realizar consultas y obtener resultados entre diferentes repositorios, no es más que una interfaz de programación que debe ser implementada y publicada como un servicio por el repositorio que sirve de fuente de datos, cabe destacar que SQL permite búsquedas asíncronas entendiéndose como el proceso de búsqueda que no se queda conectado esperando respuesta, sino que brinda un servicio de entrega de las respuestas, por lo que la fuente de la consulta debe implementar el servicio de retribución de consulta en caso de que ésta sea asíncrona, la mejor práctica para publicar este servicio es usando Servicios Web bajo el protocolo SOAP.

SQL es neutral en términos de lenguaje de consultas, además es un Application Program Interface (API) que plantea los siguientes servicios básicos: servicios de autenticación, gestión de la sesión y servicios de aplicación para la gestión de las consultas.

SQL define un conjunto de métodos que pueden ser agrupados por la funcionalidad a la que dan soporte (VanNguen, 2007):

Manejo de Consultas: métodos que permiten la configuración de los parámetros de consultas: `setQueryLanguage()`, `setResultsFormat()`, `setMaxQueryResults()`, `setMaxDuration()`.

Consultas Síncronas: los resultados de las consultas son directamente retornados por el método `synchronousQuery()`. Métodos adicionales permiten la elección de los números resultados retornados por una consulta (`setResultsSetSize()`) y además permiten conocer el número total de resultados de una consulta (`getTotalResultsCount()`).

Consultas Asíncronas: el API para este tipo de interacción define los siguientes métodos: `queryResultsListener()` para ejecutar una consulta asíncrona, `setSourceLocation()` para establecer la ubicación del cliente para que al terminar de realizar la consulta pueda entregar los resultados, `queryResultsListener()` este método forma parte de un servicio que implementa el cliente (que se convierte en servidor en este caso) para que le sean entregados los resultados.

Manejo de Sesión: define las funcionalidades para crear y destruir una sesión `createSession()`,

CreateAnonymousSession() ,DestroySeccion() respectivamente.

En **Anexo II** existe una detallada explicación de cada método de SQL.

La interfaz SQL admite lenguajes de consulta como ProLearn Query Language (PLQL), Contextual Query Language (CQL), Query Exchange Language (QEL), Very Simple Query Language (VSQL), entre otros, para la búsqueda de contenidos digitales entre repositorios, según un estudio realizado por (Fernando, 2009), el lenguaje de consulta más utilizado es el VSQL admitido por múltiples repositorios en el mundo tal es el caso de la plataforma Agrega, aunque un estudio realizado por el autor de esta investigación y reflejado en el **Anexo IV**, el lenguaje PLQL, es el lenguaje más utilizado, en las redes de OA internacionales como Globe, Ariadne y Agrega.

SQL proporciona interoperabilidad entre un ROA y aplicaciones de búsqueda y recuperación de OA, está diseñado para soportar una variedad de tecnologías de búsquedas, y lenguajes de consulta.

Existen otros protocolos que permiten implementar búsquedas federadas diferentes a SQL. A continuación se enumeran otros protocolos alternativos.

- **SRU/W** El Search/Retrieve vía URL (SRU), es un protocolo estilo REST que codifica un método de búsqueda y parámetro como una URI y retorna una instancia XML como resultado de la búsqueda. Y el Search/Retrieve Web Service (SRW) que implementa el mismo protocolo pero en una implementación SOAP (McCallum, 2006)
- **Open Knowledge Initiative (OKI)** El Open Knowledge Initiative (OKI) por sus siglas en inglés especifica la Open Service Interface Definitions (OSID) en la que se definen servicios tales como autenticación, autorización, repositorio, planificación, gestión de cursos. A diferencia del resto de iniciativas, las búsquedas no se basan en APIS, y usan interfaces que pueden tener diferentes implementaciones, haciendo la plataforma OSID dependiente. Se puede implementar tanto interacción con estado o sin estado (Sarasa, 2009)

La especificación está formada por dieciocho OSIDs, aunque no es necesario implementar todas las interfaces o los métodos que propone cada interfaz, algunas de las más importantes son: (osid_authentication, osid_authorization, osid_repository, osid_logging, osid_messaging, osid_configuration, osid_locale, entre otras)

En la **Tabla 1** se hace una comparación de SQL con otros estándares y especificaciones similares que permitan realizar las búsquedas federadas:

Tabla 1: *Tabla comparativa de SQI con otras especificaciones (Sarasa, 2009)*

Características	SQI	SRW/SRU	OKI
Síncrono	Sí	Sí	Sí
Asíncrono	Sí	No	No
Soporte para múltiples lenguajes de consulta	Sí	No	Sí
Soporte para múltiples esquemas de metadatos	Sí	Sí	Sí
Soporte para múltiples bindings	Sí, SOAP, REST, JAVA, JMS	Sí, pero limitado	Sí, JAVA y PHP
Plataforma independiente	SOAP, sí	Sí	No

Simple Publishing Interface (SPI)

El Simple Publishing Interface (SPI) por sus siglas en inglés es un protocolo que permite publicar objetos de aprendizaje y metadatos en un ROA. Para ello se define una API formada por unos conjuntos de método independientes de la tecnología lo que facilita la interoperabilidad semántica de las implementaciones de los mismos. Todos los métodos que ofrece la API requieren la creación de una sesión en el destino en el que se va a publicar el OA. Así mismo la fuente que publica el objeto, siempre usa un parámetro targetSessionID para identificar la sesión (Ternier, y otros, 2003)

Los métodos pueden ser de dos tipos:

1. Los orientados a la publicación de objetos de aprendizaje (setDataFormat, setSourceLocation, submitResourceByValue, submitResourceByReference, notifyRetrievalStatus, deleteResource).
2. Los orientados a la publicación de metadatos (setMetadataSchema, submitMetadataRecord, deleteMetadataRecord, associate, dissociate).

SPI es un protocolo que permite la publicación de OA en un repositorio, estableciendo de esta forma una comunicación con las HA y los LMS y estandariza por tanto la comunicación entre estas herramientas.

Open Archives Initiative – Protocol for Metadata Harvesting (OAI-PMH)

OAI es una iniciativa para desarrollar y promover estándares de interoperabilidad para la difusión de cualquier contenido digital. La idea básica que fomenta OAI es crear una forma simple y sencilla de intercambiar información (concretamente metadatos) entre repositorios heterogéneos que alberguen cualquier objeto que contenga metadatos asociados. OAI desarrolló el Protocol for Metadata Harvesting (PMH), por sus siglas en inglés, para permitir el intercambio de estos metadatos entre repositorios (Sarasa, 2009)

El protocolo OAI-PMH divide el mundo en proveedores de datos y proveedores de servicios. Los proveedores de datos albergan los recursos de los cuales se publican y exponen los metadatos, para ser recuperados por los proveedores de servicios, entidades que recolectan metadatos y los utilizan para dar servicios de alto nivel sobre dichos datos, por ejemplo buscadores.

El protocolo OAI-PMH proporciona una opción simple basada en el protocolo HTTP (las peticiones que se pueden realizar son operaciones GET y POST), sobre una arquitectura REST y como formato para la respuesta XML, ver **Anexos V**, donde se explican detalladamente cada uno de estos métodos.

Los metadatos pueden estar en cualquier formato que sea acordado por la comunidad correspondiente, aunque toma como base el esquema proporcionado por (DC). Es importante notar que OAI-PMH no proporciona una búsqueda a través de los metadatos, simplemente permitir tener todos los metadatos juntos en un solo lugar (Sarasa, 2009)

Además se puede restringir los resultados mediante la recolección selectiva, en las cuales la respuesta, puede ser filtrada por un rango de fecha o por la colección a la cual pertenece, o cualquier combinación de estos.

Open Archives Initiative-Object Reuse and Exchange (OAI-ORE)

La principal alternativa a OAI-PMH es la Open Archives Initiative-Object Reuse and Exchange (OAI-ORE) por sus siglas en inglés. Esta especificación tiene como objetivo definir un estándar que permita identificar, describir e intercambiar agregaciones de recursos web (Lagoze, y otros, 2007)

Para ello es necesario establecer:

- Cómo puede un proveedor de servicios codificar y desglosar descripciones de agregaciones.
- Cómo puede un usuario/máquina descubrir e interpretar descripciones de agregaciones.

La especificación representa el dominio de una agregación a través de un grafo, formado por nodos (recursos o propiedades) y arcos (relaciones). De manera que existe un nodo (llamado “mapa de recurso”) que describe al nodo que representa a la agregación en sí misma.

Este nodo queda identificado con una URI, que da acceso vía Http, a un fichero donde en un determinado esquema de metadatos, se proporcionan todos los recursos y relaciones existentes en el grafo. Observar que cada recurso agregado puede ser asimismo una agregación, y cada agregación puede tener recursos asociados con otro tipo de relaciones (Lagoze, y otros, 2007)

1.7. Arquitecturas para repositorios

La naturaleza distribuida de las instituciones y su tendencia a administrar sus contenidos en repositorios distribuidos, hace necesario disponer de modelos de referencia para crear y gestionar estos repositorios. Así han ido apareciendo diferentes modelos de referencia que tratan de estandarizar las funcionalidades y características de un repositorio digital. En esta sección se describe brevemente los modelos de referencia IMS DRI y CODRA.

IMS Digital Repositories Interoperability (IMS-DRI)

IMS-DRI es una especificación de normas y recomendaciones, que facilita un esquema funcional de una arquitectura del sistema y de un modelo de referencia completo para la interoperabilidad de repositorios.

Esta especificación tiene como objetivo facilitar el acceso a los contenidos en los repositorios para contextos educativos, con los LMS y los LCMS, pero también con otros sistemas como los portales de búsquedas (Riley, y otros, 2003)

Esta especificación se propone para la interoperabilidad entre servicios o aplicaciones que tienen las funciones más comunes de un repositorio: buscar, exponer, coleccionar, enviar, almacenar, pedir, entregar y alertar. Entre estas funciones, se reconocen cinco combinaciones como actividades principales: Buscar/Exponer, Colectar/Exponer, Enviar/Almacenar, Pedir/Entregar y Alertar/Exponer, en el **Anexo I** se resalta en detalle su descripción y recomendaciones tecnológicas.

En la **Figura 7** se muestra la arquitectura funcional entre un sistema e-learning, los repositorios digitales y los servicios de información. En la misma figura se muestran los roles, los componentes funcionales y los servicios que definen el espacio de interacción completo, pero IMS-DRI no incluye a todas éstas, sólo se enfoca a la interoperabilidad de las funciones que se han mencionado.

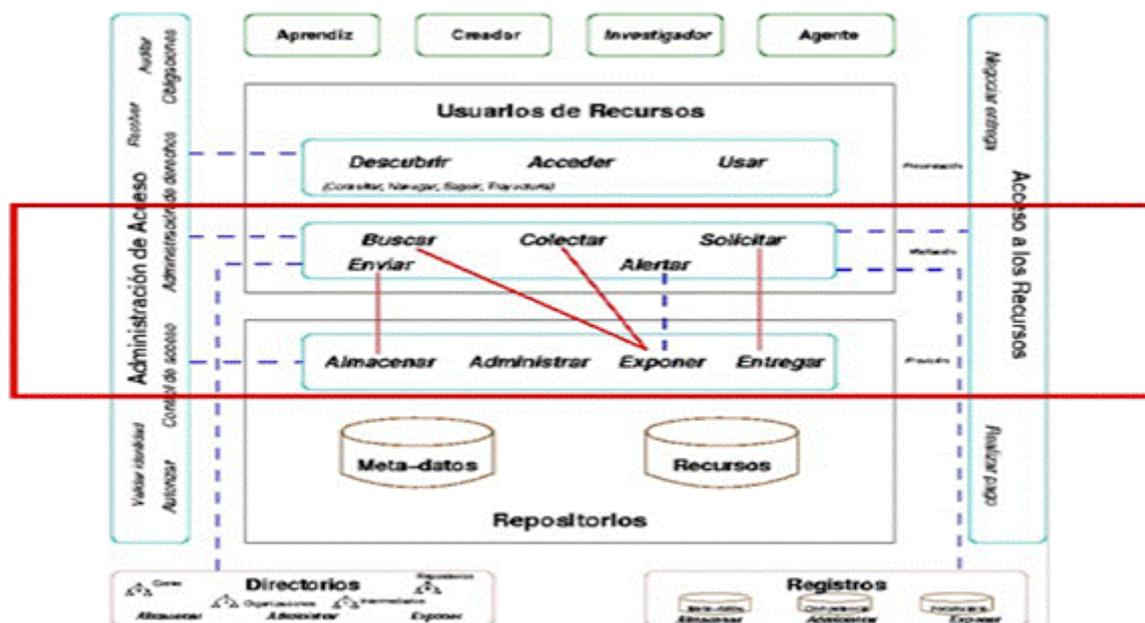


Figura 7: Arquitectura del protocolo IMS-DRI (Sarasa, 2009)

Mediante esta especificación, se consigue acceder a los contenidos almacenados en repositorios desde cualquier aplicación software que sea compatible con la especificación IMS-DRI.

Arquitectura para el Descubrimiento, Registro y Resolución de Repositorios de Objetos de Contenidos (CODRA)

La principal alternativa a IMS-DRI es Content Object Repository Discovery and Registration/Resolution Architecture, (CORDRA) por sus siglas en inglés, el cual proporciona un modelo de referencia para repositorios de objetos de aprendizaje. Define entre otras cosas cómo una federación de repositorios de objetos de aprendizaje debería ser diseñada y desplegada en la red.

En el nivel más bajo, los repositorios de contenido, publican sus metadatos en el registro de metadatos de CORDRA. CORDRA dicta el uso de un registro intermedio de registro en el cual metadatos de diferentes comunidades son agrupados, aunque no está claro qué metadatos deberían encontrarse en este registro. Lo mismo ocurre con respecto al registro principal de registros, en el nivel más alto de los registros. El registro podría contener referencias a registros intermedios, permitiendo una búsqueda federada en el nivel más alto de CORDRA o por el contrario permitir una recolección de los metadatos de todos los contenidos como si fuera un sistema centralizado (Sarasa, 2009)

Arquitectura de implementación de IMS-DRI en Agrega

En la **Figura 8** se pueden distinguir los servicios internos utilizados por el subsistema DRI, así como los tres servicios de la especificación, que pone a disposición del exterior. En este sentido para que un nodo sea interoperable y cumpla el protocolo DRI debe al menos de permitir almacenar objetos en él, buscarlos y obtener un objeto.

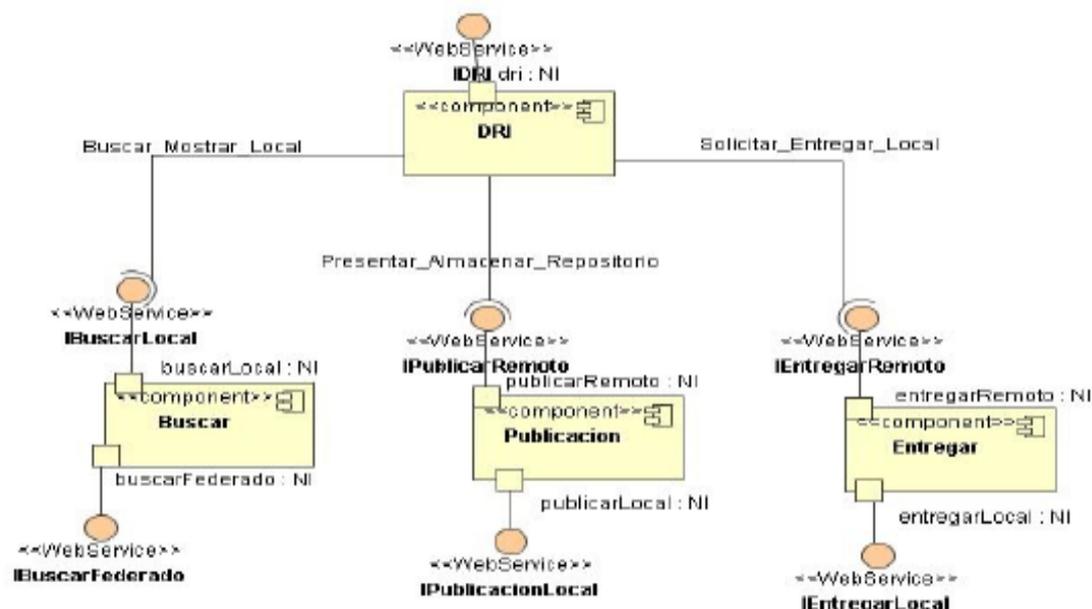


Figura 8: Arquitectura implementación de IMS-DRI (Sarasa, 2009)

Para almacenar objetos se ha definido el método `presentar_almacenar` y para recuperarlos el método `solicitar_entregar`, con el fin de facilitar la interoperabilidad y que otros repositorios o sistemas puedan integrarse con `Agrega`, todos los servicios del subsistema DRI, se han publicado al exterior en forma de servicios web (Sarasa, 2009)

1.8. Comportamiento de la interoperabilidad en repositorios del mundo y Cuba

Existen en el mundo iniciativas interesantes para lograr funcionalmente un repositorio interoperable, en este acápite se hará referencia a dos de las iniciativas más interesantes a nivel mundial (`EduSource` y `Agrega`), como modelos a seguir.

Proyecto `EduSource`

El proyecto `EduSource`²¹ pretende unir los repositorios canadienses de objetos de aprendizaje heterogéneos a través de la infraestructura necesaria para permitir la interoperabilidad entre ellos.

²¹ Disponible en <http://www.edusource.ca/>

Estos repositorios pertenecen a distintas comunidades, por lo que se hace un poco engorrosa la comunicación de estas aplicaciones.

De esta forma, el proyecto EduSource, crea EduSource Communication Layer (ECL) por sus siglas en inglés siguiendo las especificaciones y recomendaciones proporcionadas por IMS-DRI. Recordando las directrices que marca IMS-DRI en el desarrollo de su especificación, es necesario matizar que ECL ha tenido que tomar decisiones de implementación ante la falta de especificación al respecto.

Dentro de la definición de ECL se tienen dos grandes ámbitos. Por un lado, proporcionar las herramientas necesarias para crear un nuevo repositorio y permitir que se una a la red existente y por otro lado, proporcionar los componentes necesarios para que un repositorio ya existente pueda formar parte de la red que conforma ECL.

Proyecto Agrega

El proyecto Agrega²² (AGREGA) tiene como objetivo crear una federación de repositorios digitales con nodos instalados en cada una de las comunidades autónomas de España. El diseño de Agrega se realizó pensando en la necesidad de poder incorporar en la federación de repositorios a otras federaciones de repositorios que existen en el mundo.

Por otro lado también se tuvo en cuenta que un repositorio digital es una componente de construcción que debe ser fácil de integrar con otras herramientas u entornos, con la finalidad de crear entornos de educación online más complejos. Cada nodo de la federación se ha implementado usando Java y siguiendo un modelo basado en una Arquitectura Orientada a Servicios (SOA) donde el papel del proveedor de servicios lo interpreta el Nodo de Objetos Educativos Digitales y el papel consumidor lo interpretaran las Aplicaciones Clientes.

Además cada nodo dispone de un conjunto de componentes funcionales básicos tales como: autenticación, autorización, entre otros, y publica un conjunto de servicios web, en lo que se denomina interfaz de interoperabilidad, para facilitar la integración con otros sistemas. La arquitectura funcional del nodo está inspirada en el estándar IMS-DRI y las búsquedas de contenidos se realizan usando la especificación SQL.

Repositorios cubanos.

En Cuba estas aplicaciones se encuentran principalmente, en los centros de Educación Superior o en Institutos de investigación y se usan exclusivamente para almacenar información, aunque la Biblioteca

²² Disponible en <http://www.proyectoagrega.es/default/home.php>

Virtual de la Educación Superior (BIVES) implementa el protocolo OAI-PMH. Estos repositorios son generados por herramientas como DSpace, Fedora, E-prints, Greenstone, mencionados anteriormente.

Algunos de los ejemplos más citados son, el Repositorio Digital (Instituto de geografía Tropical) elaborado con la herramienta DSpace, la Biblioteca Virtual de las Ciencias en Cuba elaborada con la herramienta Greenstone y el Repositorio de la Facultad de Matemáticas y Computación de la U.H elaborado con la herramienta e-prints.

A continuación se hace referencia en la **Tabla 2** a los estándares y especificaciones usadas en algunos de los más importantes repositorios existentes en el mundo y en Cuba, para lograr la comunicación con otros sistemas.

Tabla 2: Estándares vs repositorios

No.	Nombre de Repositorios	Cubanos	OAI-PMH	OAI-ORE	SQI	OKI	CODRA	IMS-DRI	SPI
1	Repositorio Digital (Instituto de geografía Tropical)	X	X						
2	Biblioteca Virtual de la Educación Superior (BIVES)	X	X						
3	Red Telemática del IDICT	X	X						
4	Repositorio institucional de la UCI. (Biblioteca Digital)	X	X						
5	Biblioteca Virtual de las Ciencias en Cuba	X	X						

6	Repositorio de la Facultad de Matemáticas y Computación de la U.H	X	X						
7	EdNA online				X				
8	Ariadne		X		X				X
9	Merlot				X				
10	SMETE				X				
11	Agrega		X		X	X		X	
12	CELEBRATE		X						
13	CAREO				X			X	
14	Elena/Edutella				X				
15	eduSource		X					X	
16	Globe		X		X				X
17	Nime				X				

18	LACLO				X				
19	MACE				X				
20	FLOR				X				
21	EducaNext				X				
22	CGIAR				X				
23	LIONSHARE				X				
24	CiteSee				X				

En la tabla anterior se analizaron los principales repositorios y federaciones existentes en Cuba y el mundo, haciendo énfasis en búsquedas que se realizan entre los repositorios que componen dichas federaciones. También se analizaron las especificaciones y estándares que permiten la interoperabilidad de estos sistemas arrojando que los estándares más utilizados en las propuestas estudiadas son: IMS DRI como base de la arquitectura y los estándares OAI-PMH para la recolección de metadatos, SQL búsqueda federadas y SPI para la publicación de contenidos en dichos sistemas, estos resultados se muestran en la **Figura 9**.

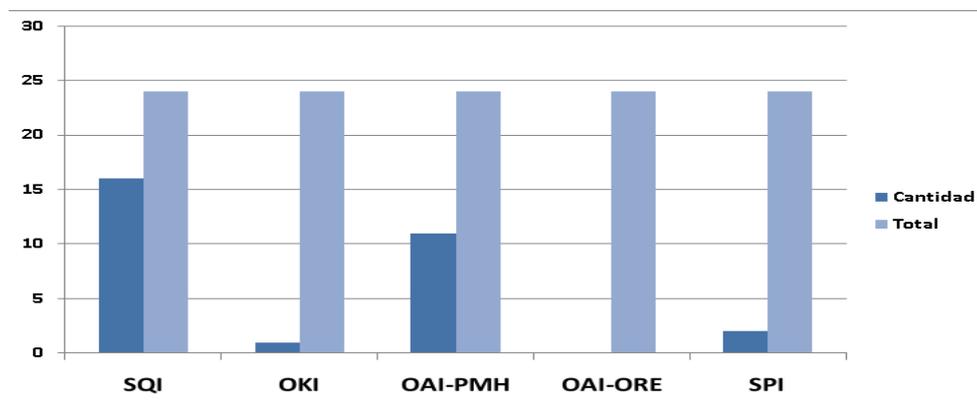


Figura 9: Utilización de los estándares en los repositorios

1.9. Servicios Web

En la actualidad se están imponiendo las arquitecturas basadas en servicios web, que facilitan el acceso a la funcionalidad de las aplicaciones a través de Internet, permitiendo acceder a servicios desde cualquier punto de la red, simplificando su utilización y facilitando la interoperabilidad entre servicios y entre aplicaciones lo que permite integrar la funcionalidad de distintas aplicaciones empresariales.

XML-Remote Procedure Call (XML-RPC)

XML-RPC es un protocolo de llamadas a procedimientos remotos que usan XML para codificar los datos, sobre http como protocolo para la transmisión de mensajes.

XML RPC define de manera sencilla como enviar el nombre de un método y una lista de argumentos de un sistema a otro. La idea fundamental de XML RPC es que un documento XML es utilizado para indicar el nombre de un método y la lista de argumentos que este necesita. Este documento XML es enviado a un servidor web utilizando HTTP POST. El servidor procesa el documento XML y ejecuta el método solicitado, retornando luego el resultado en otro documento XML. Los parámetros de procedimiento pueden ser escalares, números, cadenas, fechas, entre otras, (XML-RPC, 2011)

Representational State Transfer (REST)

La transferencia de estado representacional (REST) por sus siglas en inglés es la alternativa más simple a Simple Object Access Protocol (SOAP), es un término presentado por primera vez por Roy Fielding, para describir un estilo de arquitectura que se utiliza como modelo en los sistemas de computación Web. En la presente investigación se usará la definición:

“REST es un intento de mostrar cómo debe comportarse una aplicación Web bien diseñada: una red de páginas Web (una máquina de estados virtual) donde el usuario progresará seleccionando enlaces (transiciones de estado) que devuelven la página siguiente (el siguiente estado de la máquina) que el usuario manejará a su gusto” (Fielding, 2000)

REST define un set de principios arquitectónicos por los cuales se diseñan servicios web haciendo foco en los recursos del sistema, incluyendo cómo se accede al estado de dichos recursos y cómo se transfieren por HTTP hacia clientes escritos en diversos lenguajes.

Una implementación concreta de un servicio web REST sigue cuatro principios de diseño fundamentales:

1. Utiliza los métodos HTTP (GET, POST, PUT y DELETE) de manera explícita.
2. No mantiene estado.
3. Expone URIs con forma de directorios.
4. Transfiere XML, JavaScript Object Notation (JSON), o ambos

Existe un enorme número de aplicaciones REST en la red, tal es el caso de (Google, Facebook, Amazon.com, eBay, Yahoo!, Microsoft, entre otros). Un ejemplo de servicio Web tipo RESTful en repositorios son los provistos por el repositorio MERLOT, para la recolección de metadatos con el protocolo OAI-PMH.

Simple Object Access Protocol (SOAP²³)

SOAP es un protocolo de comunicaciones por paso de mensajes XML que es la base sobre la que sustentan los servicios web, en la mayoría de las aplicaciones empresariales. Los mensajes SOAP son unidireccionales, aunque todas las aplicaciones pueden participar como emisoras o receptoras indistintamente. Los mensajes SOAP pueden servir para muchos propósitos: petición/respuesta, mensajería asíncrona, notificación, entre otros.

Es un protocolo de alto nivel, que sólo define la estructura del mensaje y algunas reglas básicas de procesamiento de éste, siendo completamente independiente del protocolo de transporte. Ello posibilita que los mensajes SOAP puedan ser intercambiados a través de HTTP, SMTP, JMS, entre otros, donde HTTP es el más utilizado en estos momentos.

La forma general de un mensaje SOAP sería:

²³ Disponible en <http://www.w3.org/2000>

```

<ENVELOPE atribs>
<HEADER atribs><directivas /></HEADER>
<BODY atribs><cuerpo /></BODY>
<FAULT atribs><errores /></FAULT>
</ENVELOPE>

```

El significado de cada parte será:

Envelope: es el elemento raíz del formato SOAP.

- Header: es un elemento opcional, que sirve para extender las funcionalidades básicas de SOAP (seguridad, entre otros).
- Body: es el elemento que contiene los datos del mensaje. Su presencia es obligatoria.
- Fault: en caso de error, esta sección contendrá información sobre la naturaleza de éste.

Existe una especificación ampliada de SOAP llamada SwA (SOAP with Attachments) por su siglas en ingles que utiliza la codificación MIME para el transporte de información binaria.

Web Services Description Language (WSDL²⁴)

WSDL son las siglas de Web Services Description Language, un lenguaje, basado en XML que nos permite describir los servicios web que desplegamos. Además, WSDL, también se usa para localizar y ubicar dichos servicios web en la Internet.

Un documento WSDL es en realidad un documento XML que describe algunas características de un servicio web, su localización y ubicación, los métodos y parámetros que soporta y define tipos de datos a usar.

Un documento WSDL tiene, pues, una estructura similar a la siguiente:

```

<definitions><types>tipos de datos...</types>
<message>definiciones del mensaje...</message>
<portType>definiciones de operación ...</portType>
<binding>definiciones de protocolo...</binding>
</definitions>

```

Un documento WSDL puede contener además otros elementos, como extensiones, y un elemento de servicio que hace posible agrupar las diversas definiciones de varios servicios web en un sólo documento WSDL.

Universal Description, Discovery, and Integration (UDDI²⁵)

²⁴ Disponibles en <http://www.w3.org/TR/wsdl.html>

UDDI son las siglas de Universal Description, Discovery and Integration, un servicio de directorio donde las empresas y usuarios pueden publicar y buscar servicios web. UDDI es una estructura estándar e independiente de plataforma para describir estos servicios web, buscar servicios, entre otros.

UDDI está construido sobre los estándares de Internet del W3C y de la IETF (Internet Engineering Task Force), como XML, HTTP. Para describir las interfaces hacia los servicios web, utiliza el lenguaje WSDL descrito anteriormente y para las necesidades de programación multiplataforma utiliza SOAP, lo que permite una total interoperabilidad.

UDDI representa un gran avance para el desarrollo de Internet como plataforma de negocios de tecnologías de la información.

Algunos de los beneficios que se derivan del uso de UDDI son:

- Permite descubrir el negocio (o servicio) adecuado de los miles que están registrados actualmente en algunos servidores vía Internet.
- Define cómo interactuar con el servicio escogido, una vez localizado éste.
- Nos permite llegar a nuevos clientes y facilita y simplifica el acceso a los clientes ya existentes.
- Extiende el mercado de usuarios potencial de nuestros métodos de negocio y servicios.
- Describe los servicios y componentes o métodos de negocio de forma automática (o automatizable) en un entorno seguro, abierto y simple.

Extensible Markup Language (XML)

Extensible Markup Language (XML), por sus siglas en inglés, se define como: *“un conjunto de reglas que se usan para definir etiquetas semánticas las cuales organizan un documento en diferentes partes. Siendo así un meta lenguaje que define sintaxis para definir otros lenguajes etiquetados estructurados”* desarrollado por el World Wide Web Consortium (W3C).

El XML permite crear etiquetas adaptadas a las necesidades y es especialmente estricto en cuanto a lo que está permitido y lo que no, todo documento debe cumplir dos condiciones: ser válido y estar bien formado.

XML Schema nace en el 2001, procedente de los esfuerzos del W3C para paliar las deficiencias

25 Disponible en <http://www.uddi.org>

evidentes de Document Type Definition (DTD). Algunas de las principales características de XML Schema son: está definida en XML lo cual permite validar también los documentos XML Schema, permite el control y definición sobre los tipos de datos, permite describir el contenido de los documentos, facilita validar que los datos son correctos y facilita definir patrones (formatos) de datos.

Web Service Security (WS-Security, WSS)

WS-Security es una especificación de comunicación que provee los medios necesarios para aplicar seguridad a los Servicios Web. Describe las mejoras relacionadas a la mensajería SOAP y contiene especificaciones sobre cómo debe garantizarse la integridad y seguridad en mensajería de Servicios Web.

Estos mecanismos pueden ser utilizados para dar cabida a una amplia variedad de modelos de seguridad y tecnologías de encriptación.

XML Encryption

XML Encryption es una Recomendación del Consorcio Web (W3C) que especifica un proceso para cifrar datos (no únicamente documentos XML) y representar esa información cifrada a su vez en XML para que viaje por los medios de transmisión. XML Encryption, es un lenguaje cuya función principal es asegurar la confidencialidad de partes de documentos XML a través de la encriptación parcial o total del documento transportado, que puede aplicarse a cualquier recurso web, incluyendo contenido que no es XML (W3C, 2012)

XML Signature

XML Signature es una Recomendación del Consorcio Web (W3C) que provee mecanismos y especificaciones para crear firmas digitales basadas en XML. Puede aplicarse a cualquier contenido digital, incluyendo XML, y se basa principalmente en asociar claves a los datos de consulta, para garantizar a través de una firma digital la autenticidad de los mismos. Una característica fundamental es la posibilidad de firmar sólo porciones específicas del árbol XML en lugar del documento completo, además se puede firmar más de un tipo de recurso (W3C, 2012)

1.10. Metodología de desarrollo de software, tecnologías y herramientas a usar

A continuación se hará referencia a las principales tecnologías y herramientas así como la metodología de desarrollo de software a utilizar para dar cumplimiento al problema planteado en dicha investigación. Han sido utilizadas con éxito en las versiones anteriores de RHODA, por lo que su presencia queda completamente justificada.

Metodología de desarrollo de software. Rational Undefined Process (RUP)

En la creación de un software, es necesario definir una metodología de desarrollo, que no son más que procesos donde se esquematiza la evolución del proyecto, concretando **Quién hará Qué, Cuándo y Cómo**. RUP es el resultado de largos años de desarrollo, donde se unifican técnicas de desarrollo, mediante XML, y trabajo de muchas metodologías utilizadas por los clientes. RUP está caracterizado como un proceso de desarrollo iterativo e incremental, centrado en la arquitectura, y dirigido por casos de usos. (Jacobson, Ivar, y otros. 2000).

Utiliza como lenguaje de modelado UML, además define cuatro fases (conceptualización, elaboración, construcción y transición) en el desarrollo de software, divididas en nueve flujos de trabajo (modelado de negocio, requerimientos, análisis y diseño, implementación, pruebas, despliegue, configuración y administración de cambios, administración de proyecto y entorno).

Tecnologías

Los lenguajes pueden dividirse en dos grupos, los lenguajes del lado del Servidor y los lenguajes del lado del Cliente, donde ambos reconocen la filosofía de la arquitectura Cliente/Servidor para las plataformas Web. En el lado del cliente existen tecnologías como: JavaScript utilizado para ofrecer dinamismo a la aplicación, además de Cascading Style Sheets (CSS) y HyperText Markup Language (HTML), contribuyendo a la no recarga en servidores y proporcionando rapidez y optimización en los canales de comunicación. Del otro lado la principal alternativa es el procesador de hipertexto (PHP). Aunque también existen técnicas de desarrollo para la web como lo es Ajax y XML.

Javascript (JScript)

Es un lenguaje de programación interpretado con sintaxis semejante a Java y C, al igual que Java, JavaScript es orientado a objetos, pues dispone de herencia, esta se realiza siguiendo el paradigma de programación basada en prototipos, debido a que las nuevas clases se generan clonando las clases base (prototipos) y extendiendo su funcionalidad. Todos los navegadores modernos interpretan el código JavaScript integrado dentro de las páginas web. Para interactuar con una página web se provee al lenguaje JavaScript de una implementación del DOM.

Cascading Style Sheets (CSS)

CSS es un lenguaje usado para definir la presentación de un documento estructurado escrito casi siempre en HTML. La idea que se encuentra detrás del desarrollo de CSS es separar la estructura de un documento de su presentación.

Algunas ventajas de utilizar CSS son:

- Control centralizado de la presentación de un sitio web completo con lo que se agiliza de forma considerable la actualización del mismo.
- Una página puede disponer de diferentes hojas de estilo según el dispositivo que la muestre o, incluso, a elección del usuario.
- El documento HTML en sí mismo es más claro de entender y se consigue reducir considerablemente su tamaño.

HyperText Markup Language (HTML)

Se traduce al español como lenguaje de etiquetas de hipertexto. Es el lenguaje de marcado predominante para la construcción de páginas web, diseñado para estructurar textos y presentarlos en forma de hipertexto, que es el formato estándar de las páginas Web. Permite además códigos de lenguajes de programación extendiendo su capacidad y funcionalidad. Es el estándar usado en el World Wide Web. HTML ofrece los medios a través de los cuales se publica online documentos con cabeceras, texto, fotos, tablas además de incluir hojas de cálculo y otras aplicaciones directamente en los documentos.

Procesador de hipertexto (PHP)

Como lenguaje de desarrollo se propone PHP ya que este lenguaje es multiplataforma, está orientado al desarrollo de aplicaciones web manteniendo acceso a información almacenada en una base de datos, el código fuente escrito en PHP es transparente al navegador y al usuario haciendo la programación segura y confiable. Presenta capacidad de conexión con la mayoría de los motores de base de datos destacándose principalmente en MySQL y PostgreSQL.

Posee una amplia documentación en su página oficial, donde todas las funciones del sistema están explicadas y ejemplificadas en un único archivo de ayuda. Es libre, por lo que se presenta como una alternativa de fácil acceso para todos. No requiere definición de tipos de variables aunque sus variables se pueden evaluar también por el tipo que estén manejando en tiempo de ejecución. La última versión estable, 5.4.0 salió (1 de marzo de 2012)

Herramientas

Symfony

Symfony es un framework diseñado para optimizar, gracias a sus características, el desarrollo de las aplicaciones web. Se diseñó para que se ajustara a los siguientes requisitos:

- Fácil de instalar y configurar en la mayoría de plataformas (y con la garantía de que funciona correctamente en los sistemas Windows y Unix estándares).
- Sencillo de usar en la mayoría de casos, pero lo suficientemente flexible como para adaptarse a los casos más complejos.
- Código fácil de leer que incluye comentarios de phpDocumentor y que permite un mantenimiento muy sencillo.
- Fácil de extender, lo que permite su integración con librerías desarrolladas por terceros.

Symfony tiene numerosas características como lo son su publicación bajo una licencia de software libre, la amplia documentación existente, que unido a otras lo hacen un excelente framework de desarrollo.

NetBeans

NetBeans es un entorno de desarrollo libre, gratuito y sin restricciones de uso, hecho principalmente para el lenguaje de programación Java. Existe además un número importante de módulos para extender el NetBeans IDE. Es una herramienta para programadores pensada para escribir, compilar, depurar y ejecutar programas. Está escrito en Java pero puede servir para cualquier otro lenguaje de programación, como PHP5 y soporta el desarrollo de todos los tipos de aplicación Java (J2SE, web, EJB y aplicaciones móviles).

Apache

Apache es un servidor web robusto cuya implementación se realiza de forma colaborativa, además Apache es modular, de código abierto, multiplataforma, Popular (fácil conseguir ayuda/soprote).

Principales funcionalidades.

- Los módulos Apache API — se utiliza un nuevo conjunto de interfaces de programación de aplicaciones (APIs).
- Filtrado — Los módulos pueden actuar como filtros de contenido.
- Soporte a IPv6.
- Directrices simplificadas — Se han eliminado una serie de directrices complicadas y otras se han simplificado.

Respuestas a errores en diversos idiomas — Cuando usa documentos Server Side Include (SSI), las páginas de errores personalizables se pueden entregar en diversos idiomas. (Apache, 2012)

PostgreSQL

PostgreSQL es un potente servidor de base de datos relacional orientado a objetos, libre y de código abierto (open source), incluye extensa documentación, incluyendo un enorme manual y algunos ejemplos de ensayo, se ejecuta en la mayoría de los sistemas operativos más utilizados en el mundo incluyendo Linux, varias versiones de UNIX y en Windows.

A continuación otras características del gestor PostgreSQL:

1. Cuenta con comunidades muy activas y varias en español.
2. Es altamente adaptable a las necesidades del cliente.
3. Presenta un soporte nativo para los lenguajes más populares del medio: PHP, C, Perl, Python.
4. Soporta todas las características de una base de datos profesional (triggers, store procedures, funciones, secuencias, relaciones, reglas, tipos de datos definidos por usuarios).
5. Además permite realizar transacciones, subselects, disparadores, vistas, claves foráneas.

eXist-db

Exist-db es un SGBD de código abierto construido usando tecnología XML. Almacena datos en XML de acuerdo al modelo de dato de ese mismo formato. Soporta múltiples estándares de tecnologías (web principalmente) haciéndolo una excelente plataforma para el desarrollo de aplicaciones basadas en la web (Exist, 2009). A continuación se mencionan algunos de estos estándares soportados por Exist-db:

- Xquery 1.0 / Xpath 2.0 / XSLT 1.0 ó XSLT 2.0.
- Interfaces HTTP: REST, WebDav, SOAP, XMLRPC, Atom Publishing Protocol.
- Base de datos XML: XMLDB, XUpdate, XQuery.

Conclusiones parciales

Después de un análisis del estado del arte referente a la temática que se estudia se puede afirmar que:

- Se realizó una detallada explicación de las principales aplicaciones de un entorno e-learning concluyendo que es vital que las mismas puedan comunicarse e intercambiar información.

- Se propuso un nuevo modelo para lograr la interoperabilidad en los sistemas de información (publicación), el cual permitirá la publicación de OA en un repositorio, que junto a otros modelos como (colección, recolección y federación) permitirá lograr un modelo global más general de la interoperabilidad.
- En la investigación se encontraron como principales estándares para la comunicación entre estos sistemas (SQI, OAI-PMH y SPI), como lenguaje de consulta en las búsquedas federadas PLQL, como modelos globales para la arquitectura IMS-DRI, ejemplo muy claro de este modelo lo constituyen el repositorio Agrega y EduSource referentes mundiales.
- Se definieron las tecnologías y herramientas a usar estas son (Javascript, html, css del lado del cliente, php del lado del servidor, xml para describir los recursos, NetBeans IDE de programación, apache como servidor web, Postgres como base de datos relacional, eXist como base de datos nativa xml, Symfony 1.3 como Framework de desarrollo y RUP como metodología de desarrollo), se tomó como base tecnológica las usadas en las versiones anteriores de RHODA.

CAPÍTULO 2

Introducción

En este capítulo se proponen los estándares a usar, para el desarrollo de la solución propuesta, se describe la arquitectura del módulo de interoperabilidad, la estructura del sistema y la manera en que se agrupan sus elementos fundamentales, se explican las principales clases y se describe las funcionalidades arquitectónicamente más importantes. Se aborda acerca del método de Caja Negra, para la validación de la propuesta, teniendo en cuenta que la realización de las pruebas es un elemento crítico para garantizar la calidad del software.

2.1. Modelo integral de interoperabilidad en un repositorio que gestione contenidos educativos

Como se explicó en el capítulo 1, los investigadores de la (NSDL) estadounidense identificaron tres modelos de trabajo para lograr la interoperabilidad en sistemas que gestionen información, ellos son: (colección, federación y recolección). El autor de la presente investigación considera además que para lograr un modelo integral de interoperabilidad en un repositorio que gestione contenidos educativos, específicamente un ROA, es necesario que se tenga en cuenta un nuevo modelo (publicación), el cual permitirá junto con los otros modelos el mayor grado de interoperabilidad posible en un repositorio.

Estos modelos fueron analizados anteriormente de forma general, concluyendo que su utilización es de extrema importancia, a continuación se detallan cada uno y se propone la tecnología a utilizar, que en cada caso fue explicada detalladamente en el capítulo anterior.

El modelo **Colección** es un enfoque utilizado en los motores de búsquedas como Google y Yahoo, consiste en identificar y copiar los contenidos de base de datos y páginas web, hacia una base de datos centralizada donde se indexa el contenido para su posterior reutilización. Provee un mínimo de interoperabilidad en sistemas formalmente no cooperantes, a costa de un elevado consumo de recursos, por lo que el autor de la presente investigación considera que no es de vital importancia y además plantea que con la realización de los otros modelos se logra un alto grado de interoperabilidad en un ROA.

El modelo **Recolección** permitirá que los repositorios, basado en el acuerdo de un conjunto de sistemas, permita compartir determinados servicios para el intercambio simple y eficiente de información (concretamente metadatos). La recolección de metadatos es una solución muy utilizada en bibliotecas digitales y repositorios digitales, consiste en obtener los metadatos que ofrecen los proveedores de datos y guardarlo de forma centralizada en un repositorio o proveedor de servicio, para brindar

servicios de búsquedas sobre los mismos. Para dar cumplimiento a este modelo se propone el protocolo de recolección de metadatos **OAI-PMH**.

El modelo **Federación** permitirá que los repositorios obtengan el mayor grado de interoperabilidad en las búsquedas en tiempo real. Requiere de un acuerdo entre los sistemas, para la utilización de estándares y tecnologías comunes, el lenguaje de consulta constituirá la lengua común que hablarán estos repositorios. Este modelo permitirá la realización de búsquedas federadas, se propone la utilización de la interfaz de simple consulta **SQI** y el lenguaje de consulta **PLQL**.

El modelo **Publicación** permitirá que los OA creados en otras herramientas como HA o LMS y exportados a un formato común (SCORM 1_2 o SCORM 2004) puedan ser publicados en un ROA. Este modelo permitirá que sistemas heterogéneos puedan reutilizar los OA, para ello se propone la interfaz de publicación simple **SPI**.

La **Figura 10** muestra como quedaría el modelo global de interoperabilidad en un ROA, y se propone la tecnología a utilizar en cada uno de ellos.



Figura 10: Modelo global de interoperabilidad en un repositorio y propuesta de tecnologías a utilizar

2.2. Artefactos de los flujo de trabajo de las fases de inicio y elaboración de la metodología RUP

El alcance de la presente investigación sólo engloba el flujo de Implementación y Prueba, perteneciente a la fase de Construcción que establece la metodología RUP, a este le anteceden las fases de Inicio y de Elaboración. Los artefactos generados en ambas fases durante el proceso de producción de RHODA 2.2 son los establecidos por la Dirección de CALISOFT en el Expediente de Proyecto del Proceso de Mejora de la Universidad de las Ciencias Informáticas, los cuáles fueron utilizados por el autor durante la implementación de la propuesta.

Para la comprensión de la implementación desarrollada consultar los anexos a partir del número XX donde se describen los principales artefactos dígame, especificación de requisitos funcionales y no

funcionales, diagrama de clases de uso, descripción de los casos de uso y diagrama de clases del diseño. El resto de los artefactos se encuentran registrados como parte de la documentación del proyecto.

2.3. Propuesta de infraestructura tecnológica

Para dar solución al problema de investigación se presenta la siguiente propuesta de infraestructura tecnológica. Esta consta de tres capas, la capa de servicios web, la capa de negocios y la capa de acceso a datos. A continuación se detallan cada una de ellas.

Capa de servicios web: en la misma estarán publicados todos los servicios que brindará la aplicación, para ser consumidos por un sistema externo. Estos servicios corresponden a los métodos de los estándares para el intercambio de información SQI, OAI-PMH y SPI explicados anteriormente. En esta capa se encuentran los componentes de software necesarios para brindar los servicios web (API SOAP, WDSL entre otros) y dos conectores encargados de escuchar la petición, procesarla y ofrecer una respuesta, uno para los servicios publicados por REST basados en los principales métodos del protocolo HTTP (GET, POST) y otro para los publicados por SOAP.

Capa de negocio: en esta capa se encuentra toda la información que necesita la aplicación para poder trabajar con el dominio actual, además es la encargada de realizar cálculos basados en los datos de entrada o almacenado, validando los datos provenientes de la capa de servicio, así como la ejecución de algoritmos específicos para dar respuesta a estas peticiones. La capa estará compuesta por un módulo denominado **Interoperabilidad**, es un módulo más de la arquitectura de RHODA, el cual está compuesto por los componentes (OAI-PMH, SQI y SPI).

Capa de acceso a datos: es la encargada de las tareas de representación de los datos manejados por la aplicación. Esta capa manejará los datos almacenados en las base de dato relacional Postgres SQL y la base de datos XML eXist, para ello se utilizarán los lenguajes de consulta SQL para base de datos relacional apoyados por el Object Relational Mapping (ORM) Propel por sus siglas en inglés y Xquery para la base de datos nativa XML.

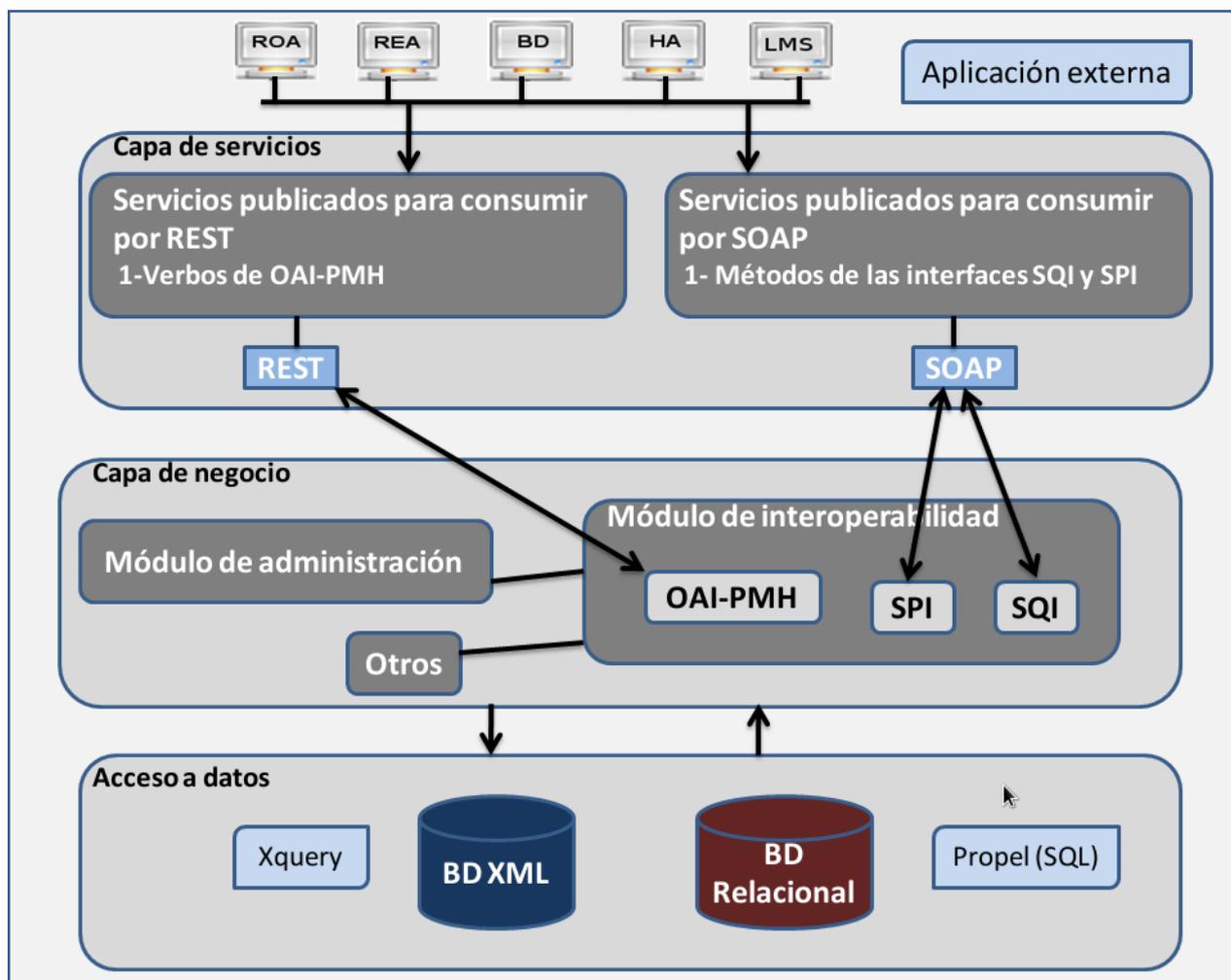


Figura 11: Infraestructura tecnológica

Descripción del módulo Interoperabilidad.

El módulo de interoperabilidad constará de cuatro componentes el primero estará dedicado a la publicación de OA, el segundo encargado de crear una sesión para poder acceder a los servicios publicados, el tercero estará dedicado a la recolección de metadatos y el último componente estará dedicado a la búsqueda federada. Para la gestión de las configuraciones de las funcionalidades es necesario que el usuario este registrado con anterioridad en el sistema y este autenticado como administrador. Dichas configuraciones se realizarán en el panel de administrador.

2.4. Estándares de codificación

Con el objetivo de lograr que el equipo de desarrollo comprenda el código de los demás desarrolladores, es necesario que se establezca un estándar de codificación. Para el desarrollo del

módulo de interoperabilidad se usará las pautas trazadas en las versiones anteriores de RHODA que a continuación serán mencionadas.

Idioma

Para nombrar variables, funciones, clases y demás se usarán nombres descriptivos y en idioma inglés.

Identificadores

Los identificadores se comienzan a escribir con minúscula y en el caso de componerse de más de una palabra se escriben una a continuación de la otra, comenzando con mayúscula a partir de la segunda.

Como se puede apreciar en el fragmento de código que describe a continuación:

```
class Item{

private $str ;

private $actionPerform ;
```

Documentación del código fuente

Una de las mejores prácticas para generar la documentación de los módulos es que esté auto contenida en los mismos, para ello en este trabajo, debido a que se utiliza PHP como lenguaje de programación, se utilizó el estándar phpDoc, el cual la mayoría de los Entornos de Desarrollo Integrado que dan soporte a PHP lo interpretan y usan para ayuda al programador, además de que ofrece la gran ventaja de poder generar páginas HTML con documentación del código fuente.

```
/**

* This method return an object with the SCORM schema for

* @return stdClass

*/

private static function getItems( $path ){
```

Identación

Para indentar cualquier fragmento de código se utilizarán espacios en blanco y nunca tabulaciones, las estructuras anidadas usarán 4 espacios para la indentación, además los lexemas que se usan para comenzar y cerrar bloques “{” y “}” no poseerán indentación alguna.

```
public function preExecute()
```

```

{
  $id=$this->getRequestParameter('id');
  if ($id == 1)
  {
    $this->redirect404 ();
  }
}

```

2.5. Modelo de Implementación

El modelo de implementación denota la programación de la aplicación en términos de componentes y subsistemas de implementación. Describe la organización de los datos, archivos, ejecutables, código fuente y directorios de acuerdo a los mecanismos de estructuración disponibles en el entorno de implementación.

Diagrama de componentes

El diagrama de componentes representa cómo un sistema de software es dividido en componentes y muestra las dependencias entre estos componentes. Los componentes físicos incluyen archivos, módulos, ejecutables, o paquetes.

El diagrama de componentes del módulo “Interoperabilidad” **Figura 12** muestra los diferentes componentes del módulo y su relación. Los estereotipos utilizados en el diagrama de componentes son:

- <<file>>: son ficheros de configuración o código fuente.
- <<framework>>: representa un marco de trabajo, que automatiza patrones para realizar tareas comunes.
- <<library>>: representa una librería estática o dinámica.
- <<database>>: representa una base de datos.
- <<executable>>: representa un fichero binario ejecutable.
- <<script>>: representa un fichero con un código fuente de un lenguaje interpretado.
- <<table>>: representa una tabla de la base de datos.

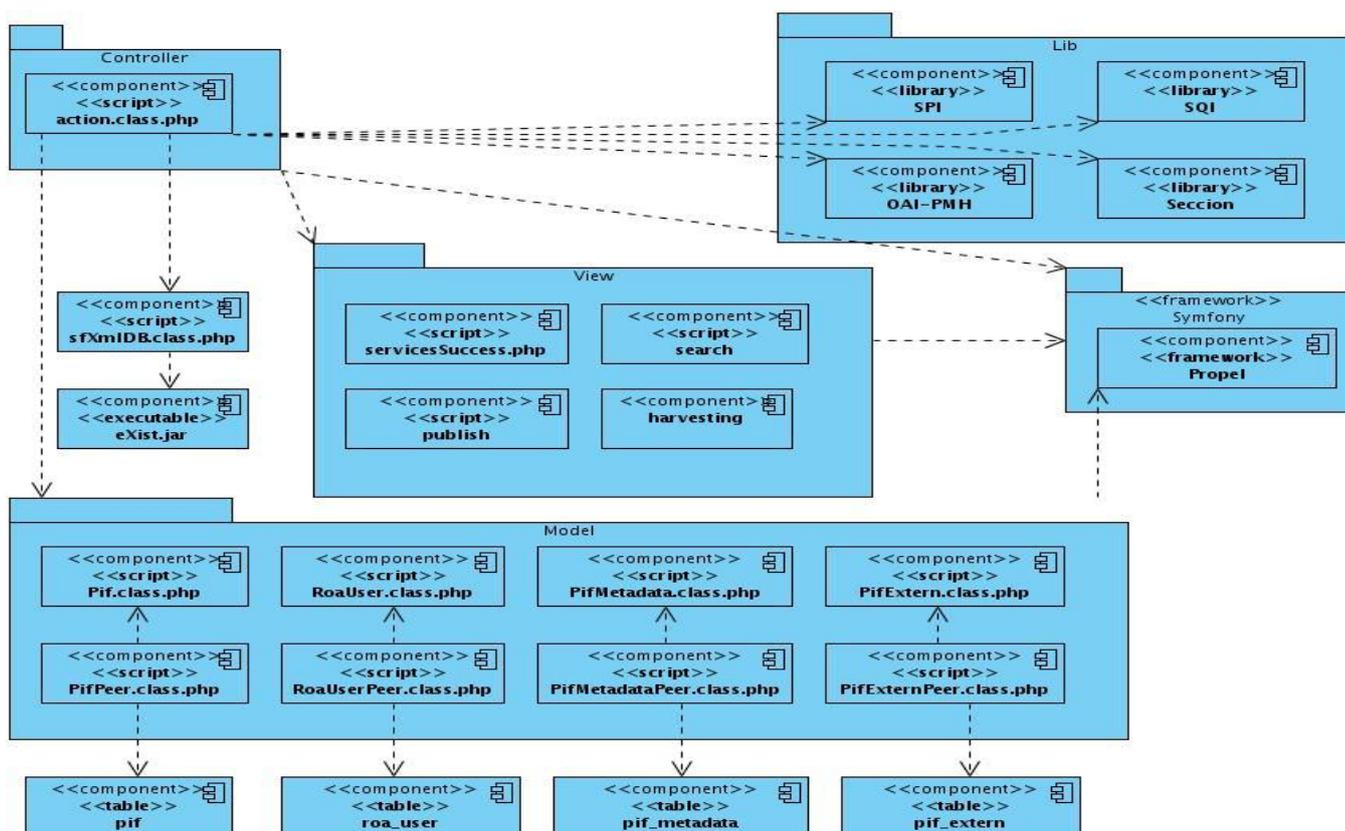


Figura 12: Diagrama de componentes del módulo de interoperabilidad

El módulo “Interoperabilidad” es uno de los que tiene mayor relevancia en el repositorio, es el responsable de que se puedan publicar OA provenientes de sistemas externos como HA y los LMC estas funcionales serán realizadas por el componente SPI, además permite que se puedan recolectar los metadatos de recursos almacenados en otros repositorios, para su posterior utilización, de la misma forma se brindan 6 servicios por el componente OAI-PMH que permiten que un sistema exterior pueda recolectar los metadatos de los OA almacenados en RHODA, además se brindan los servicios que permiten hacer búsquedas federadas sobre los OA que están almacenados en RHODA, el componente SQL realizará estas funciones.

Las funciones antes mencionadas serán publicadas como servicios web para ser consumidos por aplicaciones externas, a continuación se mencionaran los principales servicios por componente:

Servicios componente SPI

1. **setDataFormat (\$sessionId, \$setdataFormatID)** (Este servicio permite indicar el tipo de objetos que se va a almacenar en un repositorio. Dispone del parámetro setDataFormatID, que establece el formato de datos para objetos compuestos como los paquetes SCORM, AICC o

IMS-QTI, el repositorio soporta SCORM 1.2 y SCORM 1.3)

2. **setSourceLocation (\$sessionId, \$sourceLocation)** (Este servicio se usa antes de que un objeto de aprendizaje sea enviado en modo por referencia. Dispone del parámetro `sourceLocation` que permite resolver la localización de la fuente del método `notifyRetrievalStatus` y así enviar un mensaje de reconocimiento)
3. **submitResourceByReference(\$sessionId, \$reference)** (Este servicio permite publicar un OA en el repositorio por referencia, el parámetro **reference** posee la referencia del OA en el sistema externo)
4. **submitResourceByValue(\$sessionId, \$data)** (Este servicio permite publicar un OA en el repositorio por valor, se almacenera y se retornará un identificador que corresponde a dicho OA)
5. **setMetadataSchema(\$sessionId, \$metadataSchema)** (Este servicio permite a la fuente controlar el esquema de los metadatos que será publicado en el destino, y al destino le permite validar las instancias de metadatos, actualmente RHODA soporta LOM-ES como esquema de metadatos)
6. **submitMetadataRecord (\$sessionId, \$metadataRecord)** (Este servicio permite enviar para publicar una instancia de metadatos. El repositorio genera un identificador que será retornado y el cual identificará la instancia de metadatos)
7. **associate(\$sessionId, \$resourceIdentifier, \$metadataIdentifier)** (Este servicio permite asociar una instancia de metadatos con un OA publicado por referencia)
8. **dissociate(\$sessionId, \$resourceIdentifier, \$metadataIdentifier)** (Este servicio permite disociar una instancia de metadatos con un OA publicado por referencia)

Servicios componente SQI

1. **setQueryLanguage(\$targetSessionID, \$queryLanguageID)** (Este servicio permite a la fuente controlar la sintaxis usada en las instrucciones de la consulta por identificación del lenguaje de consulta, el repositorio soportara el lenguaje de consulta PLQL)
2. **setMaxQueryResults (\$targetSessionID, \$maxQueryResults)** (Este servicio define el número máximo que una consulta puede producir, especificándose como un porcentaje. El parámetro `maxQueryResults` debe ser cero o más)

3. **setMaxDuration(\$targetSessionID, \$maxDuration)** (Este servicio permite a la fuente configurar un tiempo de espera en caso de que la consulta opera en una interface de consultas asíncronas. El parámetro que controla el tiempo de espera se expresa en milisegundos)
4. **setResultsFormat(\$targetSessionID, \$resultsFormat)** (Este servicio permite a la fuente controlar el formato de los resultados retornados por el destino, el repositorio soportara LOM-ES como formato para devolver los resultados)
5. **setResultsSetSize(\$targetSessionID, \$resultsSetSize)** (Este servicio define el máximo número de resultados, que por defecto son 25. Si toma el valor de cero, entonces se entiende que quiere recuperar todos los resultados)
6. **synchronousQuery(\$targetSessionID, \$queryStatement)** (Este servicio envía una consulta al destino a través del parámetro queryStatement. Dentro de una sesión identificada por el targetSessionID. El método retorna un conjunto de instancias de metadatos acordes con la consulta realizada, queryStatement será una consulta en PLQL)
7. **GetTotalResultsCount(\$targetSessionID, \$queryStatement)** (Este servicio retorna el número total de resultados existentes de una consulta)

Servicios componente Sesión

1. **createSession(\$userID, \$password)** (Este servicio permite crear una sesión asociada a un usuario, para lo cual requiere del usuario y la contraseña como parámetros, retorna un identificador de sesión si el usuario es autenticado)
2. **createAnonymousSession()** (Este servicio permite crear una sesión no asociada a un usuario, y puede ser usado en un sistema que permite a todo las personas crear consultas)
3. **destroySession(\$sessionID)** (Este servicio permite a la fuente que inició la sesión finalizar la sesión, para lo que es necesario eliminar el identificador de sesión sessionID)

Servicios componente OAI-PMH

1. **GetRecord** (Este servicio se utiliza para recuperar un registro de metadatos individuales de un repositorio. Es necesario especificar el identificador del elemento del que se solicita el registro y el formato de los metadatos que deben incluirse en el registro)

Parámetros:

- metadataPrefix (requerido)

- identifier (requerido)

Ej. http://roa_dev.php/interoperability/request?verb=GetRecord&metadataPrefix=oai_dc&identifier=oai:Rhoda_1587

2. **Identify** (Este servicio se utiliza para recuperar información sobre un repositorio)

Ej. http://roa_dev.php/interoperability/request?verb=Identify

3. **ListRecords** (Este servicio se utiliza para recolectar los metadatos asociados a los registros almacenados en el repositorio. Los argumentos opcionales permiten la búsqueda selectiva de los registros basados en la colección a la que pertenecen y especificando la fecha de creación del mismo)

Parámetros:

- metadataPrefix (requerido)
- from (opcional)
- until (opcional)
- set (opcional)
- resumptionToken (exclusivo)

Ej. http://roa_dev.php/interoperability/request?verb=ListRecords&metadataPrefix=oai_dc&from=2011-02-28T09:07:21&until=2011-01-22T09:07:21Z&identifier=oai:Rhoda_1587

4. **ListIdentifiers** (Este servicio es una abreviatura del verbo ListRecords, devuelve solo las cabeceras de los registros, se pueden realizar búsquedas selectivas especificando la colección y la fecha en que se publicó el registro)

Parámetros:

- metadataPrefix (requerido)
- from (opcional)
- until (opcional)
- set (opcional)
- resumptionToken (exclusivo)

Ej. http://roa_dev.php/interoperability/request?verb=ListIdentifiers&metadataPrefix=oai_dc&from=2011-02-28T09:07:21&until=2011-01-22T09:07:21Z&identifier=oai:Rhoda_1587

5. **ListSets** (Este servicio se utiliza para devolver la estructura de las colecciones existentes en el repositorio)

Parámetros:

- resumptionToken (exclusivo)

Ej. http://roa_dev.php/interoperability/request?verb=ListSets

6. **ListMetadataFormats** (Este servicio se utiliza para devolver un listado de los metadatos soportados en el repositorio, se puede conocer opcionalmente el formato de un recurso especificando su identificador)

Parámetros:

- identifier (opcional)

Ej. http://roa_dev.php/interoperability/request?verb=ListMetadataFormats&identifier=oai:Rhoda_1587

2.6. Interoperabilidad entre herramientas de un entorno e-learning según la propuesta

El desarrollo del módulo interoperabilidad, permite que el repositorio de recursos educativos abiertos RHODA, pueda comunicarse con otras aplicaciones de un entorno e-learning, ver **Figura 13**. Esta comunicación permite que se puedan recolectar y realizar búsquedas, sobre los metadatos, así como reutilización y entrega de los recursos, almacenados en el sistema. Como se puede apreciar la interoperabilidad es la única alternativa viable para lograr los procesos durante el ciclo de vida de un recurso desde su creación en una herramienta de autor hasta su utilización en el LMS.

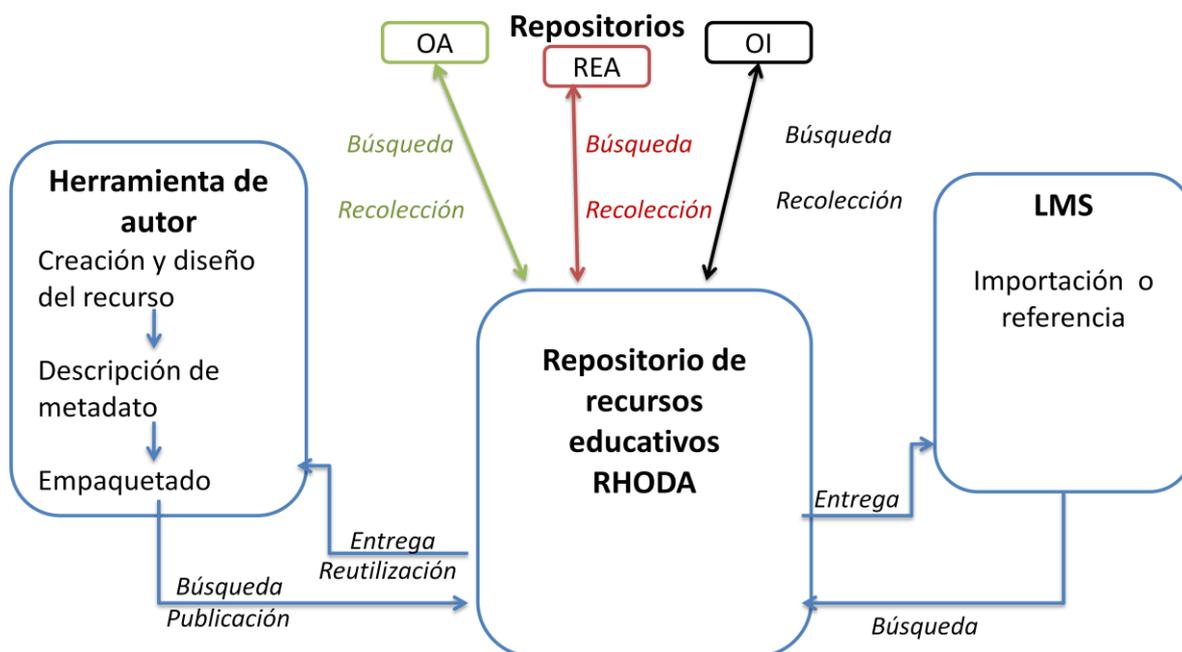


Figura 13: Interoperabilidad entre herramientas de un entorno e-learning

2.7. Pruebas de Caja Negra

Las pruebas de Caja Negra se realizan sobre las interfaces de usuario, su objetivo es garantizar que las funcionalidades sean operativas. Se les denomina así porque no se tiene conocimiento de cómo funciona, ni cuál es la estructura interna de la funcionalidad que está en prueba. Se aplican sobre la interfaz de la aplicación observando la respuesta del sistema ante determinadas acciones. Estas pruebas se realizan mediante casos de prueba, los cuales tienen como objetivo principal demostrar que las funcionalidades son correctas.

Dentro del método de caja negra se hace uso de la técnica “Partición Equivalente” por ser una de las más efectivas para examinar los valores válidos e inválidos de las entradas existentes en el software. La Partición Equivalente divide el dominio de entrada de un programa en clases de datos a partir de las cuales se derivan casos de prueba (Pressman, 2002)

A continuación se muestran algunos de los casos de prueba utilizados para validar la solución propuesta.

2.8. Diseños de caso de prueba

Módulo config

Caso de prueba: Gestionar Repositorio para la recolección de metadatos (OAI-PMH)

Descripción: El caso de uso (CU) inicia cuando el administrador decide adicionar, modificar o eliminar un repositorio digital.

Condiciones de ejecución: El usuario tiene que estar autenticado en la aplicación y debe tener el rol de administrador.

Tabla 3: Escenarios para el caso de prueba "Gestionar repositorio para la recolección de metadatos"

Nombre de la sección	Escenarios de la sección	Descripción de la funcionalidad
EC 1: Adicionar un repositorio.	EC 1.1 Insertar un repositorio.	Se inserta el repositorio.
EC 2: Eliminar un repositorio.	EC 2.1 Eliminar un repositorio.	
	EC 2.2 Se selecciona Aceptar.	Se elimina el repositorio deseado.
EC 3: Editar un repositorio.	EC 3.1 Modificar los datos de un repositorio.	Se modifica los datos de un repositorio.
EC 4: Visualizar formatos de catalogación de un repositorio.	EC 4. 1 Visualizar formatos de catalogación soportados por el repositorio.	Se visualiza todos los formatos soportados por el repositorio.
EC 5: Visualizar correos de los administradores del repositorio a recolectar.	EC 5.1 Visualizar los correos de los administradores.	Se visualizan todos los correos de los administradores de los repositorios a recolectar.
EC 6: Listar todos los repositorios	EC 6.1 Listar todos los repositorios, para la recolección de metadatos.	Se listarán todos los repositorios, para la recolección de metadatos.

Caso de prueba Adicionar un repositorio.

Tabla 4: Caso de prueba "Adicionar un repositorio"

Escenario	Descripción	Tipo	Periodicidad	Descripción	Horario de actualización	Uri	Respuesta del sistema	Flujo central
EC 1	Se selecciona la opción Adicionar nuevo repositorio						Brinda la posibilidad de introducir los datos siguientes: <ol style="list-style-type: none"> 1. URL base del repositorio. 2. Tipo de repositorio. 3. Descripción del repositorio. 4. Horario de 	<ol style="list-style-type: none"> 1. Seleccionar menú administrar. 2. Seleccionar opción repositorio "Gestionar recolección de repositorio (OAI-PMH)" 3. Seleccionar opción "Adicionar nuevo repositorio"

							<p>actualización.</p> <p>5. Periodicidad de la actualización.</p> <p>Insertar un repositorio.</p> <p>Cancelar la operación en cualquier momento.</p>	
EC 1.1	Adicionar repositorio	V	V	V	V	V	<p>Extrae algunos datos de los servicios Identify y ListMetadataFormats del repositorio.</p> <p>Valida los datos.</p> <p>Guarda el nuevo repositorio.</p> <p>Se activa la interfaz Listar repositorio para la recolección.</p>	<ol style="list-style-type: none"> 1. Seleccionar menú administrar. 2. Seleccionar opción "Gestionar recolección de repositorio (OAI-PMH)" 3. Seleccionar opción "Adicionar nuevo repositorio" 4. Presiona el botón Adicionar
EC 1.2		I	V	V	V	V	<p>Se muestra un mensaje "Campo obligatorio en blanco" y además se muestra en rojo dicho componente.</p>	<ol style="list-style-type: none"> 1. Seleccionar menú administrar. 2. Seleccionar opción "Gestionar recolección de repositorio (OAI-PMH)" 3. Seleccionar opción "Adicionar nuevo repositorio"
		V	I	V	V	V		
		V	V	I	V	V		
		V	V	V	I	V		
		V	V	V	V	I		
EC 1.3	Selecciona la opción cancelar.						<p>Se cierra la interfaz para insertar el repositorio.</p> <p>Se activa la interfaz Listar repositorio para la recolección.</p>	<ol style="list-style-type: none"> 1. Seleccionar menú administrar. 2. Seleccionar opción "Gestionar recolección de repositorio (OAI-PMH)" 3. Seleccionar opción "Adicionar nuevo repositorio" 4. Presiona el botón Cancelar

Caso de prueba eliminar un repositorio.

Tabla 5: *Caso de prueba "Eliminar repositorio"*

Escenario	Descripción	Respuesta del sistema	Flujo central
EC 2	Se selecciona la opción Eliminar repositorio	Muestra un mensaje de advertencia para confirmar la eliminación de un repositorio.	<ol style="list-style-type: none"> 1. Seleccionar menú administrar. 2. Seleccionar opción repositorio "Gestionar recolección de repositorio (OAI-PMH)" 3. Seleccionar opción "Eliminar repositorio"

EC 2.1	Se selecciona la opción Aceptar	El repositorio es eliminado.	<ol style="list-style-type: none"> 1. Seleccionar menú administrar. 2. Seleccionar opción repositorio “Gestionar recolección de repositorio (OAI-PMH)” 3. Seleccionar opción “Eliminar repositorio” 4. Seleccionar opción “Aceptar”
--------	---------------------------------	------------------------------	-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Caso de prueba Editar un repositorio.

Tabla 6: Caso de prueba “Editar la información de un repositorio”

Escenario	Descripción	Tipo	Periodicidad	Descripción	Horario de actualización	Uri	Respuesta del sistema	Flujo central
EC 3	Se selecciona la opción Editar						Brinda la posibilidad de modificar los datos siguientes: <ol style="list-style-type: none"> 1. URL base del repositorio. 2. Tipo de repositorio. 3. Descripción del repositorio. 4. Horario de actualización. 5. Periodicidad de la actualización. Aceptar. Cancelar la operación en cualquier momento.	<ol style="list-style-type: none"> 1. Seleccionar menú administrar. 2. Seleccionar opción repositorio “Gestionar recolección de repositorio (OAI-PMH)” 3. Seleccionar opción “Editar”
EC 3.1	Editar repositorio	V	V	V	V	V	Valida los datos. Guarda los datos modificados. Se activa la interfaz Listar repositorio para la recolección.	<ol style="list-style-type: none"> 1. Seleccionar menú administrar. 2. Seleccionar opción “Gestionar recolección de repositorio (OAI-PMH)” 3. Seleccionar opción “Editar” 4. Presiona el botón Aceptar
EC 3.2		I	V	V	V	V	Se muestra un mensaje “Campo obligatorio en blanco” y además se muestra en rojo dicho	<ol style="list-style-type: none"> 1. Seleccionar menú administrar. 2. Seleccionar opción “Gestionar recolección
		V	I	V	V	V		
		V	V	I	V	V		

		V	V	V	I	V	componente.	de repositorio (OAI-PMH) 3. Seleccionar opción "Editar"
		V	V	V	V	I		
EC 3.3	Selecciona la opción cancelar.						Se cierra la interfaz para Editar los datos del repositorio. Se activa la interfaz Listar repositorio para la recolección.	1. Seleccionar menú administrar. 2. Seleccionar opción "Gestionar recolección de repositorio (OAI-PMH)" 3. Seleccionar opción "Editar" 4. Presiona el botón Cancelar

Caso de prueba Visualizar formatos de catalogación de un repositorio.

Tabla 7: Caso de prueba "Visualizar formatos de catalogación de un repositorio"

Escenario	Descripción	Respuesta del sistema	Flujo central
EC 4	Se selecciona la opción visualizar formatos de catalogación soportados por el repositorio.	Brinda la posibilidad de conocer los formatos de catalogación soportados por el repositorio a cual se le van a recolectar los metadatos.	1. Seleccionar menú administrar. 2. Seleccionar opción repositorio "Gestionar recolección de repositorio (OAI-PMH)" 3. Seleccionar opción "Lenguajes de catalogación"

Caso de prueba Visualizar formatos e-mail de los administradores.

Tabla 8: Caso de prueba "Visualizar e-mail de los administradores"

Escenario	Descripción	Respuesta del sistema	Flujo central
EC 5	Se selecciona la opción visualizar e-mail.	Brinda la posibilidad de conocer el e-mail de los administradores del repositorio.	1. Seleccionar menú administrar. 2. Seleccionar opción repositorio "Gestionar recolección de repositorio (OAI-PMH)" 3. Seleccionar opción "e-mail"

Caso de prueba Listar repositorios para la recolección de metadatos.

Tabla 9: Caso de prueba “Listar repositorios para la recolección de metadatos”

Escenario	Descripción	Respuesta del sistema	Flujo central
EC 6	Se listan todos los repositorios almacenados.	Brinda la posibilidad de listar todos los repositorios.	1. Seleccionar menú administrar. 2. Seleccionar opción repositorio “Gestionar recolección de repositorio (OAI-PMH)”

Módulo de interoperabilidad

Caso de prueba: Recolectar de metadatos de OA en RHODA.

Descripción: El caso de uso (CU) inicia cuando se consume el servicio ListRecords por una aplicación externa.

Condiciones de ejecución: Los valores de los parámetros set, from, until, metadataPrefix y resumptionToken, deben ser los especificados por el verbo Identify y el protocolo OAI-PMH.

Caso de prueba Recolección de metadatos de OA en RHODA.

Tabla 10: Caso de prueba" Recolectión de metadatos de OA en RHODA"

Escenario	Descripción						Respuesta del sistema	Flujo central
		set	from	metadat aPrefix	until	resumpt ionToke n		
EC 1	Se consume el servicio recolectar metadatos de los OA en RHODA						Brinda todos los metadatos de los OA solicitados, por los parámetros (set, from, until, metadataPrefix y resumptionToken), en el formato definido por el protocolo OAI-PMH.	Servicios publicados por la aplicación, en el módulo interoperability, acción request.
EC 1.2		I	V	V	V	I	Se muestra el mensaje de error correspondiente, definido por el protocolo OAI-PMH.	Servicios publicados por la aplicación, en el módulo interoperability, acción request, servicio ListRecords.
		V	I	V	V	I		
		V	V	I	V	I		
		V	V	V	I	I		
		I	I	I	I	V		

Clasificación de las no conformidades

La descripción de las No conformidades ayuda a la acelerada gestión de los proyectos y a que el proceso de pruebas de liberación se realice en un tiempo reducido y planificado, sin atender contra el

cronograma de desarrollo y los compromisos con los clientes.

Se puede realizar una evaluación más profunda del software, se obtienen mejores estadísticas de cuáles son las No Conformidades más comunes en los diferentes tipos de software y de darle un mejor tratamiento en el proceso de liberación del software.

Una No Conformidad es un fallo en el Sistema de gestión de la Calidad que puede producirse por varias razones: no alcanzar el nivel de aceptación establecido en un determinado *indicador* y errores en la documentación del Sistema. Se trata de una desviación entre lo que hay escrito (lo que hemos dicho que vamos a hacer) y lo que ha ocurrido (lo que hemos hecho). Este fallo queda registrado en un informe y se establecen *las acciones preventivas y correctivas* necesarias para arreglar lo que no funcione y evitar que vuelva a ocurrir.

Las mismas se clasifican de acuerdo al nivel de importancia en:

1. **Las No Conformidades Significativas:** Son aquellas que afectan la calidad del producto o servicio de manera visible, impidiendo o no el cumplimiento de algún requisito.
2. **Las No Conformidades No Significativas:** Son aquellas que resultan menos visibles, que no atentan el cumplimiento de algún requisito.
3. **Las Recomendaciones:** Son aquellas que quedan en función de la apreciación del probador para oportunidades de mejoras del producto o servicio.

Las No Conformidades se clasifican además en función de sus características de acuerdo al tipo de artefacto como:

1. **No Conformidades de Documentación:**
 - Formato.
 - Error técnico.
 - Otros errores.
 - Correspondencia con otra documentación.
2. **No Conformidades de Aplicación:**
 - Validación.
 - Opciones que no funcionan.

- Errores de interfaz.
- Funcionalidad.
- Excepciones.
- Correspondencia de lo implementado con lo documentado.

3. No Conformidades comunes (para ambos artefactos):

- Ortografía.
- Redacción.
- Errores de idioma.

Resultados de las pruebas de caja negra

Los resultados obtenidos durante las iteraciones fueron descritos en la **Tabla 11**. En la misma puede ser apreciadas la cantidad de no conformidades (27), de las cuales once fueron de validación, cinco Errores de interfaz, seis excepciones (**No Conformidades de Aplicación**), tres de ortografía y dos fueron de redacción, las que fueron solucionadas (cerradas)

Tabla 11: Resultados de las pruebas funcionales

Iteración	No conformidades	Cerradas	No proceden
1ra	16	16	0
2da	8	8	0
3ra	3	3	0
4ta	0	-	-

Conclusiones Parciales

Después del estudio, desarrollo y validación de la propuesta se puede afirmar que:

- Los estándares para lograr la interoperabilidad que más se utilizan en este tipo de sistema son SQI, OAI-PMH y SPI, para los modelos federación, recolección y publicación respectivamente.
- Para la implementación de los modelos, se usó la arquitectura REST y el protocolo SOAP.
- La propuesta abarca hasta el nivel tres (interoperabilidad estructural), definidos en el capítulo 1.
- El desarrollo del módulo de interoperabilidad en RHODA, permite que la aplicación pueda comunicarse e intercambiar recursos educativos u OA con otras herramientas de un entorno e-learning.

- Se validó la propuesta con el método de prueba Caja Negra, para eliminar las no conformidades, presentadas por la aplicación.

CONCLUSIONES

Luego de terminar el presente trabajo se arribó a las siguientes conclusiones:

1. El estudio del arte permitió identificar los principales estándares usados para el intercambio de información de los repositorios digitales con otras aplicaciones de un entorno e-learning.
2. Se adicionó un nuevo modelo (publicación) al modelo establecido por los investigadores de NCDL estadounidense, el cual permite la publicación de OA en RHODA, desde diferentes herramientas para su posterior reutilización.
3. Se desarrolló una propuesta tecnológica en tres capas, que permite que se puedan publicar los servicios, para ser consumidos por una aplicación externa.
4. Las pruebas permitieron validar la solución propuesta arrojando resultados positivos.

Por todo lo anteriormente planteado se concluye que los objetivos propuestos para el presente trabajo han sido cumplidos satisfactoriamente. Se incluyen algunas recomendaciones para futuros trabajos que den continuidad a la presente investigación.

RECOMENDACIONES

Las recomendaciones para la presente investigación que pueden servir de guía para la continuidad de la misma son:

1. Implementar algunas funcionalidades de las especificaciones Web Service Security (WSS) para la seguridad de los servicios web de las interfaces SQI y SPI.
2. Integrar el módulo de interoperabilidad con la librería traductor de estándares, actualmente en pruebas funcionales, que permitirá la reutilización de recursos descritos en otros formatos diferentes a IEEE LOM.

BIBLIOGRAFÍA

1. **AGIMO. 2012.** Australian Government Information Management Office. *Interoperability Technical Framework for the Australian Governmen*. [En línea] 2012. [Citado el: 13 de 01 de 2012.] <http://www.agimo.gov.au/publications/2005/04/agtifv2>.
2. **Apache. 2012.** Apache Software Foundation. [En línea] 2012. [Citado el: 23 de 01 de 2012.] <http://www.apache.org/>.
3. **Atom. 2012.** [En línea] 2012. [Citado el: 13 de 01 de 2012.] 3. <http://tools.ietf.org/html/rfc5023>.
4. **Baltazar, Manjón, Sierra Rodríguez, José Luis y Moreno, Pablo. 2006.** *Uso de estándares aplicados a Tic*. 2006.
5. **Barité, M. 2000.** *Diccionario de Organización y Representación del Conocimiento. KODictionary. KODictionary*. 2000.
6. **Blanco, Libro. 2010.** Libro Blanco de la Universidad Digital. 2010.
7. **Borgman, L C. 1999.** *What are Digital Libraries, Who is Building Them, and Why?* . Zagreb : s.n., 1999.
8. **Calzada Prada, Francisco Javier. 2010.** Repositorios, bibliotecas digitales y CRAI. Madrid, España : s.n., 2010.
9. **CANARIE. 2001.** White Paper for a Learning Object Repository. [En línea] 2001. [Citado el: 04 de 03 de 2005.] http://oknl.edu.gov.on.ca/eng/pdf/1_3_13_1.pdf.
10. **CARRODEGUAS, Y. 2008.** *Análisis y Diseño de una herramienta de autor Web Interoperable*. 2008.
11. **Chiarani, Marcela Cristina, Guadalupe Pianucci, Irma y Leguizamón, Guillermo. 2006.** [En línea] 2006. [Citado el: 17 de 02 de 2011.] http://www.dirinfo.unsl.edu.ar/~profeso/PagProy/articulos/CACIC_2006-repositorio.pdf.
12. **Downes, S. 2004.** The Learning Marketplace. Meaning, Metadata and Content Syndication in the Learning Object Economy. [En línea] 2004. [Citado el: 12 de 01 de 2012.] <http://www.downes.ca/files/book3.htm>.
13. **Eap, T. 2003.** The communication middleware for collaborative learning object repository network. PhD thesis, Simon Fraser University, Irvine. 2003.
14. **ECL.** eduSource Communication Layer. Interoperability for digital repositories. [En línea] [Citado el: 29 de 11 de 2011.] <http://ecl.iat.sfu.ca>.
15. **ELENA.** ELENA Project. [En línea] [Citado el: 05 de 12 de 2011.] <http://www.elena-project.org/>.
16. **Exist. 2009.** Exist. [En línea] 2009. [Citado el: 12 de 01 de 2012.] <http://exist.sourceforge.net/>.

17. **Fielding, Roy T. 2000.** Architectural Styles and the Design of Network-based Software Architectures. Universidad de California : s.n., 2000.
18. **Fuente, Gema Bueno de la. 2008.** *ANÁLISIS DE LA INTEROPERABILIDAD ENTRE LOS SISTEMAS DE APOYO A LA FORMACIÓN DE TECMINHO.* 2008.
19. —. **2010.** Modelo de repositorio institucional de contenido educativo (RICE): la gestión de materiales digitales de docencia y aprendizaje en la biblioteca universitaria 2010 . UNIVERSIDAD CARLOS III DE MADRID DEPARTAMENTO DE BIBLIOTECONOMÍA Y DOCUMENTACIÓN : s.n., 2010.
20. **Gil González, Ana, de la Prieta, Fernando y M. C, Juan.** Sistema multiagente orientado a la búsqueda, recuperación y filtrado de objetos digitales educativos. Dept. de Informática y Automática Universidad de Salamanca : s.n.
21. **Gómez, Laureano Felipe. 2007.** *INTEROPERABILIDAD EN LOS SISTEMAS DE INFORMACIÓN DOCUMENTAL.* 2007.
22. **Hatala, M., Richards, G., Eap, T., and Willms, J. 2004.** The edusource communication language: implementing open network for learning repositories and services. In SAC '04: Proceedings of the 2004 ACM symposium on Applied computing, pages 957_962, New York, NY, USA. ACM Press. 2004.
23. **IBM. 2009.** Rational Rose Enterprise. [En línea] 2009. [Citado el: 12 de 01 de 2012.] <http://www-142.ibm.com/software/products/es/es/enterprise>.
24. **IDABC EIF. 2008.** Draft document as basis for EIF 2.0. [En línea] 2008.
25. —. **2004.** European Interoperability Framework for pan-European eGovernment Services. [En línea] 2004. [Citado el: 23 de 01 de 2012.] <http://ec.europa.eu/idabc/servlets/Doc?id=19529>. Consultado: 13-04-2010].
26. **IDABC. 2004.** European Interoperability Framework for pan-European eGovernment Services. Versión 1.0. 2004. [En línea] 2004. <http://europa.eu.int/idabc/en/document/3761..>
27. **IEEE. 1990.** *IEEE Standard Computer Dictionary: A Compilation of IEEE Standard Computer Glossaries.* 1990.
28. **IFWG. 2005.** INTEROPERABILITY FRAMEWORK WORKING GROUP. *Australian Government Technical Interoperability Framework (AGTIF) [en línea]. Version 2. Commonwealth of.* [En línea] 2005. [Citado el: 16 de 01 de 2012.] http://www.finance.gov.au/publications/australian-government-technical-interoperability-framework/docs/AGTIF_V2_-_FINAL.pdf.
29. **IMS. 2012.** IMS Global Learning Consortium From Innovation To Impact! [En línea] 2012. [Citado el: 19 de 01 de 2012.] <http://www.imsglobal.org/content/packaging>.

30. **Jacobson, Ivar. 2000.** *El Proceso Unificado de Desarrollo de Software*. 2000.
31. **JORUM+Project. 2004.** The JISC Online Repository for [learning and teaching] Materials. [En línea] 2004. [Citado el: 11 de 05 de 2005.] 24. http://www.jorum.ac.uk/docs/Vol1_Fin.pdf.
32. **Khare, R., & Taylor, R. N. 2004.** Extending the Representational State Transfer (REST). 2004.
33. **Krsulovic, Ernesto. 2012.** Blog de la Web Semántica. [En línea] 2012. [Citado el: 28 de 04 de 2012.] <http://www.dcc.uchile.cl/~ekrsulov/prj/ws-blog/>.
34. **Lagoze, Car y De Sompel, Herbert van. 2007.** Compound information objects: the OAI/ORE Perspective. [En línea] 2007. [Citado el: 14 de 07 de 2007.] <http://www.openarchives.org/ore/documents/CompoundObjects200705.html>.
35. **López Guzmán, Clara. 2005.** *Los Repositorios de Objetos de Aprendizaje como soporte a un entorno e-learning*. Salamanca : s.n., 2005.
36. **López, A. 2007.** Guía para la puesta en marcha de un repositori institucional. [En línea] 2007. [Citado el: 22 de 10 de 2011.] <http://espacio.uned.es/fez/eserv.php?pid=bibliuned:469&dsID=presentacionA>.
37. **McCallum, S. H. 2006.** A look at new information retrieval protocols: SRU, OpenSearch/a9, CQL, and XQuery. In World Library and Information. Congress: 72nd IFLA General Conference and Council. IFLA : s.n., 2006.
38. **Montilva, Jonás, Orjuela, Ailin y Rojas, Mauri. 2010.** [En línea] 2010. [Citado el: 02 de 11 de 2011.] 27. http://www.minas.medellin.unal.edu.co/index.php?option=com_docman&task=doc_download&gid=1634&Itemid=285.
39. **Paradigm, Visual. 2012.** Visual Parading Official Site. [En línea] 2012. [Citado el: 14 de 01 de 2012.] <http://www.visual-paradigm.com/>.
40. **Porter, D. 2002.** A Report on Learning Object Repositories. [En línea] 2002. [Citado el: 03 de 05 de 2005.] <http://www.canarie.ca/funding/elearning/lor.pdf>.
41. **PostgreSQL. 2012.** PostgreSQL. [En línea] 2012. [Citado el: 14 de 01 de 2012.] <http://www.postgresql.org/>.
42. **Potencier, Fabien y Zaninotto, François. 2008.** *Symfony La guía definitiva*. 2008.
43. **Pressman, Roger. 2002.** *Ingeniería de Software, un enfoque práctico*. 2002.
44. **Ramos, Orapma. 2007.** EPrints y la creación de repositorios virtuales. Estudio de un caso. La biblioteca virtual de la Facultad de Matemática y Computación de la Universidad de La Habana. [En línea] 2007. [Citado el: 11 de 11 de 2009.] http://bvs.sld.cu/revistas/aci/vol16_2_07/aci07807.html.

45. **RDF. 2012.** RDF. [En línea] 2012. [Citado el: 25 de 01 de 2012.] <http://www.w3.org/RDF/>.
46. **Real Academia Española.** Real Academia Española. Diccionario de la Lengua Española. 22a Edición. [En línea] [Citado el: 10 de 12 de 2011.] <http://buscon.rae.es/>.
47. **Red TTnet España. 2006.** La Formación Sin Distancia. 2006.
48. **Riley, K y Mckell, M. 2003.** IMS Digital Repositories Interoperability-Core Functions Information Model. [En línea] 2003. [Citado el: 15 de 01 de 2012.] http://www.imsglobal.org/digitalrepositories/driv1p0/imsdri_infov1p0.html.
49. **Rodríguez Gairín, Josep Manuel y Sulé Duesa, Andre. 2009.** DSpace: un manual específico para gestores de la información y la documentación. [En línea] 2009. [Citado el: 28 de 10 de 2011.] <http://www.ub.es/bid/20rodri2.html>.
50. **Rodriguez, Leonardo. 2010.** Análisis y Diseño de la versión 3.0 de RHODA. 2010.
51. **S. Ternier, D.Massart, F.** A simple publishing interface for learning object repositories.
52. **Sarasa, Antonio. 2009.** Diseño de repositorios digitales interoperables. 2009.
53. **SHETH, A. P. 1999.** Changing Focus on Interoperability in Information Systems: from system, syntax, structure to semantics. 1999.
54. **Smith, N., y otros. 2008.** Proceedings of World Conference on Educational Multimedia, Hypermedia and Telecommunications. [En línea] 2008. [Citado el: 16 de 02 de 2012.] <http://www.editlib.org/p/28625>.
55. **SSL. 2011.** Secure Socket Layer (SSL). [En línea] 2011. [Citado el: 12 de 03 de 2012.] <http://www.iec.csic.es/criptonomicon/ssl.html>.
56. **Ternier, S. y Duval, E. 2003.** Web services for the ARIADNE knowledge pool system. ARIADNE Foundation. : s.n., 2003.
57. **VanNguen, Nhu. 2007.** Binding the Simple Query Interface. 2007.
58. **W3C. 2012.** Sitio oficial de la W3C. [En línea] 2012. [Citado el: 03 de 02 de 2012.] <http://www.w3.org/Signature/>.
59. **—. 2012.** Sitio Oficial de W3C. [En línea] 2012. [Citado el: 11 de 03 de 2011.] <http://www.w3.org/TR/xmlenc-core/>.
60. **XML-RPC. 2011.** What is XML-RPC? [En línea] 2011. [Citado el: 22 de 11 de 2011.]