



CONSIDERACIONES PARA UNA ARQUITECTURA INTEROPERABLE DE REPOSITORIOS DE OBJETOS DE APRENDIZAJE

Tesista

Ing. Jhonatan Forero

Directores

Dr. Alejandro Hossian (UTN-UNLa) y Mg. Darío Rodríguez (UNLa)

TRABAJO PRESENTADO PARA OBTENER EL GRADO
DE
ESPECIALISTA EN INGENIERIA DE SISTEMAS DE INFORMACION

**ESCUELA DE POSGRADO
FACULTAD REGIONAL BUENOS AIRES
UNIVERSIDAD TECNOLOGICA NACIONAL**

JUNIO, 2013

RESUMEN

La evolución de las tecnologías de información, ha permitido que se adopten en los campos educativos y empresariales, la creación de repositorios distribuidos de objetos de aprendizaje, para la compartición, reutilización y localización de los recursos que se requieren, para instruir o capacitar dentro del contexto de educación virtual. Siendo la interoperabilidad un factor limitante para su implementación, porque depende de un alto grado de acoplamiento de los componentes involucrados, un proceso de adhesión de repositorios que es complejo técnicamente, y una falta de seguridad en el canal de comunicación, lo que hace necesaria, una arquitectura que solucione la interoperabilidad en plataformas heterogéneas teniendo en cuenta los problemas mencionados. En el presente trabajo, se propone analizar una matriz comparativa de las diferentes arquitecturas que existen, para luego proponer una serie de consideraciones, que generan la propuesta de una arquitectura híbrida, que permite interoperar repositorios distribuidos de una manera flexible y segura en ambientes heterogéneos.

ABSTRACT

The evolution of information technologies has allowed to be taken in the education and business fields, the creation of distributed repositories of learning objects, for sharing, reuse and location of the resources required, for instruction or training in the virtual education context. Interoperability being a limiting factor for implementation, because it depends on a high degree of coupling of the components involved, an adhesion process repository is technically complex, and a lack of security in the communication channel, becomes necessary, an architecture that solves interoperability across heterogeneous platforms considering the above problems. In this paper, we propose to analyze a matrix comparing the different architectures that exist, and then propose a set of considerations, which generate a proposed hybrid architecture that allows distributed repositories interoperate in a flexible and safe in heterogeneous environments.

ÍNDICE

1. INTRODUCCIÓN	1
2. ESTADO DE LA CUESTIÓN	3
2.1. Arquitectura orientada a servicios	3
2.2. Arquitectura Orientada a Objetos.	5
2.2.1. Corba	6
2.3. Arquitectura Orientada a Componentes	7
2.3.1. Dcom	7
2.4. Arquitectura Orientada a Recursos	8
2.4.1. Rest	9
2.5. Objetos de Aprendizaje	10
2.5.1. Metadatos	11
2.6. Repositorio de Objetos de Aprendizaje	12
2.7. Matriz Comparativa	13
2.8. Consideraciones de la Comparación	13
3. DESCRIPCIÓN DEL PROBLEMA	17
4. SOLUCIÓN	19
4.1. Descripción de la solución	19
4.2. Discusión sobre la viabilidad de la solución	21
5. CONCLUSIONES	23
6. REFERENCIAS	25

ÍNDICE DE FIGURAS

Figura 4.1	Componentes de la Arquitectura Híbrida (AHROA)	20
------------	--	----

ÍNDICE DE TABLAS

Tabla 2.1	Matriz comparativa de los diferentes estilos arquitectónicos.....	14
-----------	---	----

1. INTRODUCCION

En este capítulo se presenta el contexto del trabajo final de especialidad (sección 1.1) y su estructura (sección 1.2).

1.1. CONTEXTO DEL TRABAJO FINAL DE ESPECIALIDAD

Actualmente la evolución de las tecnologías de información ha provocado que se creen o mejoren estándares para la construcción de aplicaciones distribuidas. El crecimiento de la web y la necesidad de compartir información con fines académicos y empresariales en forma de recursos digitales llamados objetos de aprendizaje, ha creado la necesidad de construir repositorios de aprendizaje distribuidos que permitan reutilizar los objetos para la generación de contenidos sin importar su ubicación geográfica o tecnología utilizada.

Según la IEEE, un Objeto de Aprendizaje puede considerarse como una entidad digital con características de diseño instruccional, que puede ser usado, reutilizado o referenciado durante el aprendizaje soportado en computador; con el objetivo de generar conocimientos, habilidades, actitudes y competencias en función de las necesidades del alumno. (IEEE,2002).

Los contenidos de un objeto de aprendizaje deben ser aprovechados en distintos tiempos y contextos, por lo que el objetivo principal de una implementación, es la disponibilidad de los repositorios de objetos, su publicación y recuperación en ambientes heterogéneos.

Un ROA (Repositorio de Objetos de Aprendizaje) es una colección de recursos (objetos o unidades de aprendizaje) que son accesibles a través de una red. El objetivo de un repositorio es facilitar la reutilización de recursos educativos, facilitando el acceso a los recursos almacenados en el mismo.

Las funciones que debe proveer un ROA según la ADL son:

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

Debido a la necesidad de diseñar e implementar Repositorios de aprendizaje distribuidos (ROA) de forma eficiente, surgen desafíos como la interoperabilidad, reusabilidad y el rendimiento, a raíz de estos factores ha surgido distintas definiciones o aproximaciones de cómo implementar un repositorio de aprendizaje con cada una de las diferentes arquitecturas que más se adaptan a este tipo de necesidad:

- Arquitecturas orientadas a servicios.
- Arquitecturas orientadas a objetos.
- Arquitecturas orientadas a componentes.
- Arquitecturas orientadas a recursos.

Las diferentes arquitecturas mencionadas anteriormente, implementadas de forma independiente presentan distintas ventajas y desventajas, que serán abordadas con el desarrollo del presente trabajo, donde se realizara una discusión a través de una matriz comparativa, lo que dará origen a la propuesta de una arquitectura híbrida (AHROA) que permita interoperar repositorios de objetos de aprendizaje distribuidos en ambiente heterogéneos de forma segura y flexible.

2. ESTRUCTURA DEL TRABAJO FINAL DE ESPECIALIDAD

En este capítulo Introducción se presenta el contexto del trabajo final de especialidad y su estructura.

En el capítulo Estado del Arte se presenta el estado de la cuestión sobre las diferentes arquitecturas que existen para la implementación de repositorios distribuidos. se presenta la arquitectura orientada a servicios, arquitectura orientada a objetos, la arquitectura basada en componentes, la arquitectura basada en recursos, se introducen los objetos de aprendizaje, se presentan los repositorios de objetos de aprendizaje, se formula una matriz comparativa, y se realizan consideraciones sobre la comparación.

En el capítulo Descripción del Problema se centra la problemática abordada en el trabajo de especialidad.

En el capítulo Solución Considerada se presenta la solución propuesta, y se discute su viabilidad.

En el Capítulo Conclusiones se presentan conclusiones preliminares del trabajo realizado.

2. ESTADO DEL ARTE

En este capítulo se presenta el estado de la cuestión sobre las diferentes arquitecturas que existen para la implementación de repositorios distribuidos. Se presenta la Arquitectura Orientada a Servicios (sección 2.1), Arquitectura Orientada a Objetos (sección 2.2), Arquitectura Basada en Componentes (sección 2.3), Arquitectura Basada en Recursos (sección 2.4), Objetos de Aprendizaje (sección 2.5), Repositorios de objetos de aprendizaje (sección 2.6), Matriz comparativa (sección 2.7), Consideraciones sobre la comparación (sección 2.8)

2.1. ARQUITECTURA ORIENTADA A SERVICIOS

La arquitectura Orientada a servicios, es la arquitectura más difundida en el mundo de los servicios web, es un estilo de arquitectura de software cuyo propósito es lograr un débil acoplamiento entre los componentes de software que interactúan entre sí, las bases de SOA provienen de las experiencias en la utilización de tecnologías basadas en objetos distribuidos, como pueden ser Distributed Component Object Model (DCOM) y Common Object Request Broker Architecture (CORBA).

SOA es un modelo arquitectónico de software creado y usado para diseñar modelos de negocio empaquetados como servicios [Morales, 2008], SOA es el producto de una evolución de las siguientes tecnologías:

- XML (Extensible Markup Language): Es uno de los lenguajes más utilizados para el intercambio de datos sobre la Web, El lenguaje XML está concebido para describir objetos de datos llamados Documentos XML y describir de cierta forma los programas que los procesan. Está restringido bajo la norma ISO 8879 el Estándar Generalized Markup Language. XML es un lenguaje etiquetado, característica que le permite definir objetos de datos estructurados en partes bien definidas llamadas elementos.
- RPC (Remote Procedure Call): Es un protocolo que permite ejecutar una rutina en un equipo o segmento de red de manera remota.
- Web Services: Un Servicio Web es una aplicación de software identificada por un URI (Uniform Resource Identifier), cuyas interfaces se pueden definir, describir y descubrir mediante documentos XML. Los Servicios Web hacen posible la interacción entre agentes

de software (aplicaciones) utilizando mensajes XML. Intercambiados mediante protocolos de Internet.[World Wide Web Consortium [W3C]], en general son servicios utilizados para transmitir y recibir datos por aplicaciones heterogéneas vía web, los servicios web aportan interoperabilidad entre las aplicaciones mediante el uso de XML, SOAP, WSDL y UDDI sobre los protocolos de la Internet. XML es usado para describir los datos, SOAP se ocupa para la transferencia de los datos, WSDL se emplea para describir los servicios disponibles y UDDI se ocupa para conocer cuáles son los servicios disponibles.

SOA proporciona una metodología y un marco de trabajo basado en servicios, el objetivo es proveer la reutilización de un determinado servicio evitando el desarrollo de interfaces, lo cual permite proveer ventaja en la construcción de sistemas distribuidos.

Una Arquitectura Orientada a Servicio está compuesta por elementos funcionales y elementos relacionados con la calidad de servicio [Salvador, 2006].

Elementos funcionales:

- Transporte: para llevar las peticiones de servicios y respuestas entre el proveedor y el consumidor del servicio.
- Protocolo de comunicación del servicio: se establece entre el proveedor y el consumidor del servicio.
- Descripción del servicio: describe servicio, cómo debe invocarse y que datos son requeridos para la invocación.
- Servicio: Describe un servicio que está disponible para utilizarse.
- Proceso de negocio: conjunto de servicios, invocados de una manera específica, con una determinada secuencia y con ciertas reglas particulares para llevar a cabo la funcionalidad de negocio requerida.
- Registro de Servicio: repositorio de servicios y las descripciones de las mismas.

Aspectos de la calidad del servicio:

- Política: reglas o condiciones entre el proveedor y consumidor de los servicios.
- Seguridad: Conjunto de reglas aplicados para la identificación, autorización y el control de acceso de los consumidores de servicios.
- Transacción: Conjunto de atributos que pueden ser aplicados a un grupo de servicios para conseguir un resultado consistente.

- **Gestión:** Conjunto de atributos que se aplican para manejar a los proveedores del servicio o a los consumidores.

Las definiciones de SOA identifican la utilización de servicios web empleando SOAP y WSDL en su implementación, sin embargo se puede implementar SOA con cualquier tipo de tecnología que soporte servicios web, en comparación con una arquitectura orientada a objetos, SOA esta formada por servicios débilmente acoplados y altamente operables por lo que los componentes de software se vuelven altamente reutilizables.

2.2. ARQUITECTURA ORIENTADA A OBJETOS

Los componentes de este estilo son los objetos que se basan en: encapsulamiento, herencia, polimorfismo.

- La encapsulación permite al programador definir modelos que contienen tanto métodos (comportamiento) como atributos (datos) con diferentes niveles de acceso.
- La herencia permite que los datos y procedimientos de un modelo (la clase) se definan una sola vez y se reutilicen por subclases de objetos, lo cual conduce a estructuras llamadas jerarquía de clase (una clase hereda comportamiento [métodos] y estructuras de datos [atributos] desde la clase base.
- El polimorfismo, que literalmente significa "muchas formas", permite a un segmento de código de programa enviar un mensaje a un objeto sabiendo que el objeto receptor responderá correctamente aun cuando la clase precisa del objeto no es conocida. Así familias enteras de objetos pueden compartir los mismos nombres de métodos que permitan a otras aplicaciones reutilizar grandes partes (componentes) de código desarrollado.

Los objetos interactúan a través de invocaciones de funciones y procedimientos y esto es posible por la publicación de interfaces, las interfaces están separadas de las implementaciones, el mejor ejemplo de Arquitecturas orientadas a objetos es CORBA(Common Object Request Broker), en el cual las interfaces se definen mediante IDL (Interface Description Language)

2.2.1 CORBA

CORBA es un Middleware o marco de trabajo estándar y abierto de objetos distribuidos que permite a los componentes en la red interoperar en un ambiente común sin importar el lenguaje de desarrollo, sistema operacional, tipo de red, etc. En esta arquitectura, los métodos de un objeto remoto pueden ser invocados en un ambiente distribuido y heterogéneo a través de un ORB (Object Request Broker). Además del objetivo básico de ejecutar simplemente métodos en objetos remotos, CORBA adiciona un conjunto de servicios que amplían las potencialidades de éstos objetos y conforman una infraestructura sólida para el desarrollo de aplicaciones críticas de negocio.

CORBA se sustenta sobre los siguientes pilares:

- IDL (Interface Definition Language): Es un lenguaje de definición de interfaces donde el principal propósito es establecer un contrato entre cliente y servidor indicando que servicios van a estar accesibles para el cliente desde el servidor.
- ORB (Object Request Broker): se encarga de enviar las peticiones a los objetos y retornar las respuestas a los clientes que invocan las peticiones.
- GIOP (General Inter ORB Protocol) es la especificación de representación de datos y el protocolo de interoperabilidad de CORBA, define como se comunican distintos ORB.

La arquitectura desarrollada por el OMG (Object Management Group) es abierta e independiente y fue diseñada con los siguientes objetivos: orientación a objetos, transparencia de localización, independencia de un lenguaje de programación e interoperabilidad. El modelo OMA (Object Management Architecture) definido por el Object Management Group estructura los componentes de la aplicación en cuatro grandes categorías en función del nivel de reusabilidad de los mismos:

- Horizontal Facilities, utilizables como servicios completos en un amplio rango de aplicaciones.
- Common Object Services, utilizables como bloques elementales de construcción de aplicaciones ofrecen servicios preconstruidos garantizados que simplifican e desarrollo de aplicaciones complejas.
- Domain Facilities, que proporcionan componentes reutilizables en un dominio concreto de aplicación (por ejemplo procesos continuo, fabricación discreta, aviónica o sistemas médicos)

- **Application Specific Objects:** Objetos especialmente contruidos para una aplicación concreta

Para el funcionamiento de una aplicación CORBA se hace necesaria la existencia de servicios adicionales que se despliegan en forma de objetos distribuidos, en el cual se destacan el servicio de nombres, responsable de realizar la resolución de referencias a objetos y el servicio de eventos cuyo papel es llevar a cabo la gestión de los eventos entre objetos.

2.3. ARQUITECTURA ORIENTADA A COMPONENTES

Esta arquitectura se enfoca en la descomposición del diseño en componentes funcionales o lógicos que exponen interfaces de comunicación bien predefinidas.

Un modelo de componentes es un estándar que define las interfaces de los componentes y los mecanismos para interconectarlos entre ellos, las interfaces de un componente determinan tanto las operaciones que el implementa como las que precisa utilizar de otros componentes durante su ejecución, cada interface está determinada por el conjunto de atributos y métodos públicos que el componente implementa y el conjunto de eventos que emite, la interacción tradicional entre componentes está basada en llamadas a procesamiento remotos para la invocación de los métodos. Uno de los modelos más representativos de esta arquitectura es el estándar de Microsoft DCOM.

2.3.1 DCOM

Es una tecnología propietaria de Microsoft para desarrollar componentes de software distribuidos sobre varios ordenadores y que se comunican entre sí, DCOM extiende el modelo COM de Microsoft y proporciona la comunicación entre la infraestructura del servidor de aplicaciones COM+ de Microsoft. COM es un componente de software que promueve la reutilización permitiendo que aplicaciones y sistemas sean contruidos a partir de componentes binarios.

DCOM pretende solucionar los problemas más comunes que surgen en un entorno de trabajo distribuido:

- Fiabilidad ante posibles fallos en la red.
- Fiabilidad ante posibles fallos en el hardware.
- Eficiencia en términos de carga de la red.

- Posibilidad de trabajar con máquinas de diferentes prestaciones situadas en distintas áreas geográficas.

DCOM oculta la localización de los componentes, y la forma en la que el cliente se conecta a ellos y realizar una llamada a sus funciones es siempre la misma. DCOM no sólo no requiere cambios en los programas fuente, sino que tampoco es necesario recompilarlos, porque la reconfiguración cambia la forma en que los componentes se conectan entre ellos. Esto simplifica en gran medida la tarea de distribuir los diferentes componentes de la aplicación con el fin de optimizar el rendimiento. Respecto a la seguridad, DCOM implementa su propio sistema de seguridad con el fin de evitar que cada aplicación se vea obligada a desarrollar el suyo de manera independiente. DCOM proporciona seguridad a las aplicaciones distribuidas sin necesidad de incluir en el cliente o en el componente elementos de codificación o diseño específicos. Al igual que ocurría con la localización de los componentes DCOM también oculta sus necesidades de seguridad.

La cadena de protocolos de DCOM está basada en DCE RPC, para que sea fácil implementar DCOM en plataformas para las que DCE RPC ya está disponible. DCE RPC define una norma estándar para convertir estructura de datos en memoria y parámetros en los paquetes de la red. Su Representación de Datos de Red (NDR, Network Data Representation) es una plataforma neutral y proporciona un set de tipos de los datos.

DCOM emplea comunicaciones de tipo RPC para llevar a cabo las relaciones entre cliente y servidor, en el protocolo RPC para invocar una función el cliente hace una llamada al stub del cliente, el stub empaqueta los parámetros en un mensaje y se lo pasa al protocolo de comunicación para que lo lleve al servidor, en DCOM al stub del cliente se le llama proxy y al del servidor stub.

2.4. ARQUITECTURA ORIENTADA A RECURSOS

Es un estilo arquitectónico basado en REST, su principal función es exponer recursos, esta arquitectura pretende ser un método sistemático para desarrollar servicios en la Web que se adhieran a los principios de REST, la identificación de recursos aparece como un elemento esencial, un recurso consiste en una entidad que será lo suficientemente importante para ser referenciada, cada recurso debe tener al menos un único punto de acceso que lo distinga de otros, esta referencia única se llama URIs(Uniform Resource Identifier), las URIs de los recursos permiten que éstos se interrelacionen, formando una red interconectada navegable por agentes electrónicos, tal como funciona la web.

La Arquitectura orientada a recursos define cuatro propiedades que un sistema adherido a ella debe tener:

- Referencialidad (Addressability): se debe exponer la información a través de una URI.
- Carencia de estado (statelessness): cada solicitud HTTP debe realizarse de manera independiente de las otras, por lo que debe contener toda la información necesaria para ser llevada a cabo
- Conectividad (connectedness): Ésta exige que los recursos no vivan de manera aislada, sino que establezcan enlaces entre ellos, entregando al cliente los estados vecinos al actual y posibilitando así la navegación

2.4.1 REST

REST (Representational State Transfer) es un estilo arquitectónico para sistemas hipermedia distribuidos tales como la web. REST queda definido por cuatro restricciones de interfaz: identificación de recursos, manipulación de recursos mediante representaciones de estos mensajes auto-descriptivos, e hipermedia como motor del estado de la aplicación. Cabe destacar que REST no es un estándar, ya que es tan solo un estilo de arquitectura. Aunque REST no es un estándar, está basado en estándares:

- HTTP
- URL
- Representación de los recursos: XML/HTML/GIF/JPEG/, entre otros.
- Tipos MIME: text/xml, text/html, entre otros.

La finalidad de REST es exponer recursos a través de URIs y HTTP, no servicios a través de interfaces de mensajería. No debe por tanto confundirse con otros protocolos basados en RPC como SOAP O XML-RPC.

Los objetivos de REST son:

- Escalabilidad de la interacción con los componentes: La Web ha crecido exponencialmente sin degradar su rendimiento.

- Generalidad de interfaces: Gracias al protocolo HTTP, cualquier cliente puede interactuar con cualquier servidor HTTP sin ninguna configuración especial.
- Puesta en funcionamiento independiente: HTTP permite la extensibilidad mediante el uso de las cabeceras, a través de las URIs, a través de la habilidad para crear nuevos métodos y tipos de contenido.
- Compatibilidad con componentes intermedios: Los más populares intermediarios son varios tipos de proxys para Web. Algunos de ellos, las caches, se utilizan para mejorar el rendimiento. Otros permiten reforzar las políticas de seguridad: firewalls.

Para manipular los recursos, los componentes de la red (clientes y servidores) se comunican a través de una interfaz estándar (HTTP) e intercambian representaciones de estos recursos (los ficheros que se descargan y se envían). La petición puede ser tramitada por cualquier número de conectores (por ejemplo clientes, servidores, cachés, túneles, etc.) Así, una aplicación puede interactuar con un recurso conociendo el identificador del recurso y la acción requerida, no necesitando conocer si existen cachés, proxys, cortafuegos, túneles o cualquier otra cosa entre ella y el servidor que guarda la información. La aplicación, sin embargo, debe comprender el formato de la información devuelta (la representación), que es por lo general un documento HTML o XML, aunque también puede ser una imagen o cualquier otro contenido.

El manejo de los recursos se realiza utilizando las operaciones propias del protocolo HTTP:

- POST: Crea los contenidos de un URI
- GET: Recupera los contenidos
- PUT: Introduce modificaciones en un URI existente
- DELETE: Elimina un URI.

2.5. OBJETOS DE APRENDIZAJE

El concepto de objeto de aprendizaje se desarrolló a partir de la necesidad de reutilizar los materiales de aprendizaje digitales. Las tecnologías para OA ofrecen ventajas económicas lo mismo que pedagógicas. Los materiales de aprendizaje se crean sólo una vez pero se utilizan repetidamente en diferentes contextos con el objetivo de compensar el costo de producción. Además, debido a su alta calidad y cuidadoso diseño cualquier maestro, instructor o estudiante los puede utilizar

fácilmente. Aunque las ventajas para los procesos formativos parecen ser evidentes, los objetos de aprendizaje no han alcanzado hasta el momento una acogida y uso generalizados. Esto se debe en parte a que mientras la reutilización de los recursos multimedia y la no duplicación ahorran tiempo y dinero, todavía es muy significativo el esfuerzo para identificar, catalogar, almacenar, buscar, recuperar y, finalmente, volver a utilizar un objeto de aprendizaje [DUV04; DOW05; OLI05].

Poder definir lo qué es un objeto de aprendizaje es un tema altamente discutido en las comunidades relacionadas. Aunque la mayoría de investigadores podría fácilmente diferenciar lo que es y lo que no es un objeto de aprendizaje, generar consenso alrededor de su definición ha sido muy difícil. Tal vez la definición más amplia y al mismo tiempo la más criticada fue propuesta por el IEEE Learning Technologies Standard Comité LTSC [IEE02]. Dicha definición establece que un objeto de aprendizaje es cualquier entidad, digital o no digital, que puede ser utilizada, reutilizada o referenciada durante los procesos formativos soportados en las tecnologías. Si bien el objetivo de esta definición es evitar la exclusión de cualquier objeto existente, también es cierto que bajo esa definición cualquier cosa puede ser un OA [WIL02]. Una definición más concreta fue propuesta por Wiley [WIL00], según la cual un objeto de aprendizaje es cualquier recurso digital que puede ser reutilizado para apoyar un proceso formativo.

2.5.1 METADATOS

Todos los datos que se pueden utilizar para describir un objeto de aprendizaje se pueden considerar como metadatos del objeto. Según LTSC el propósito de los metadatos es facilitar la búsqueda, la evaluación, la adquisición y el uso de los OA. Por lo tanto, una definición general para metadatos es cualquier pieza de información que se pueda utilizar para buscar, evaluar, adquirir y utilizar objetos de aprendizaje. Por ejemplo, el título puede ayudar a encontrar un OA relevante; una revisión creada por un usuario puede ayudar a evaluar la pertinencia del objeto para otro usuario; el enlace a recursos reales así como la información acerca de los derechos de autor del OA puede ayudar a adquirirlo correctamente y la información técnica acerca del objeto, como tipo de archivo o tamaño, puede ayudar a seleccionar las herramientas adecuadas para utilizarlo.

Esta definición es ampliamente aceptada en la comunidad, sin embargo, existe cierto grado de confusión acerca de cómo relacionar los estándares de los metadatos de los objetos de aprendizaje. Uno de los estándares define un esquema con conjunto de campos de datos que describen al objeto: título, autor, clasificación y fecha de publicación —similar a un registro de librería—; otro podría definir el esquema con título, duración y edad estimada del público objetivo como los metadatos más importantes para almacenar. El propósito de estos esquemas es permitir la interoperabilidad

entre los diferentes sistemas que contienen metadatos. Si ambos sistemas acuerdan un estándar de metadatos común será posible el intercambio de información sobre OA, al menos en principio.

2.6. REPOSITORIO DE OBJETOS DE APRENDIZAJE

Un Learning Object Repository LOR, es cualquier sistema que almacena material de aprendizaje digital y que ofrece algún tipo de indexación y búsqueda o interface de navegación para ese material. Existen dos modelos básicos para repositorios de objetos de aprendizaje: (1) el modelo centralizado, en el que los metadatos de los OA se encuentran en un sólo servidor [DOW03]; este tipo de arquitectura se evidencia en la Figura 1, donde los metadatos se almacenan en un servidor y los contenidos se almacenan en muchos otros y (2) el modelo distribuido, en el que los metadatos de los objetos se encuentran en varios servidores conectados; este tipo de modelo utiliza una arquitectura peer-to-peer para permitir que cualquier número de servidores se comuniquen entre sí [DOW03].

Los sistemas de objetos aprendizaje de todo el mundo pueden acceder a estos metadatos para conformar su propio conjunto de recursos de aprendizaje. El LOR recuperará sólo los metadatos relevantes para cada Solicitud de búsqueda, por lo que filtra los metadatos que los sistemas de aprendizaje pueden acceder. Los OA se pueden compartir de varias maneras: pueden ser publicados en la web, hacerlos disponible en foros en línea, incluso pasar personalmente de usuario a usuario o a través LORs. En su forma más común, los repositorios almacenan el objeto y las instancias de metadatos asociadas. Estos LOR proporcionan algún tipo de indexación a la que los usuarios pueden añadir nuevos objetos junto con sus metadatos, además, proporcionan algún tipo de búsqueda o navegación para facilitar el acceso a los contenidos del repositorio. Los ejemplos más populares de repositorios de objetos de aprendizaje son:

- ARIADNE Knowledge Pool System [DUV01]. Se originó a partir de un proyecto europeo para crear un repositorio de material didáctico en la región. Se basa en una arquitectura distribuida que le permite a cada nodo mantener el control de sus propios materiales. Con más de una década de existencia, se puede considerar como uno de los más antiguos repositorios.
- Conexions [BAR07]. Es un repositorio que nació de la necesidad de compartir materiales para el procesamiento de señales digitales y que se ha extendido a otros campos. Puede ser

considerado como uno de los más nuevos y uno de los repositorios con mayores éxitos. Se basa en una licencia Creative Commons que permite la libre distribución y adaptación del material.

- Maricopa Learning Exchange. Es un pequeño repositorio que pertenece a un reducido grupo de instituciones. Su objetivo es proporcionar objetos empaquetados que se puedan reutilizar fácilmente en Learning Management Systems. La base para su contribución se limita solamente a la facultad de los Maricopa Community Colleges.

Los referatorios de objetos de aprendizaje son un importante sub-tipo de LOR que sólo almacenan metadatos mientras que el objeto se almacena en otro lugar, usualmente en un servidor Web. Los ejemplos más populares son:

- MERLOT [MAL01]. Es una iniciativa de EE.UU. para catalogar el material de aprendizaje en la web y es uno de los más antiguos. Está abierto a la contribución pero tiene un sistema en el que especialistas revisan el material en línea, al que aplican amplias evaluaciones y calificaciones. Este modelo es único en la comunidad de las LOR y continúa creciendo fuertemente.
- INTUTE. Es una iniciativa en el Reino Unido en la que un grupo de especialistas cataloga los materiales en línea. Está cerrado a contribuciones externas y trata de mantener un nivel uniforme de calidad. Debido a su edad y a la financiación continua, es uno de los mayores referatorios existentes.

2.7. MATRIZ COMPARATIVA

En la Tabla 2.1. Se presenta una matriz comparativa destacando las principales ventajas y desventajas de los diferentes estilos arquitectónicos mencionados en este capítulo.

2.8. CONSIDERACIONES DE LA COMPARACION

Con la matriz comparativa pudimos observar los puntos fuertes y débiles de las arquitecturas planteadas en este trabajo. Es evidente que las arquitecturas que ofrecen mayores beneficios son la arquitectura Orientada a Servicios SOA, y la arquitectura orientada a recursos ROA.

	ARQUITECTURA ORIENTADA A SERVICIOS	ARQUITECTURA ORIENTADA A OBJETOS	ARQUITECTURA BASADA EN COMPONENTES	ARQUITECTURA BASADA EN RECURSOS
Heterogeneidad	<ul style="list-style-type: none"> Permite integrar diferentes plataformas y Sistemas operativos por medio de servicios web 	<ul style="list-style-type: none"> CORBA provee un modelo de computación distribuida Cliente/Servidor implementada a través de un broker 	<ul style="list-style-type: none"> COM es una manera de implementar objetos neutrales con respecto al lenguaje, de manera que pueden ser usados en entornos distintos de aquel en que fueron creados. 	<ul style="list-style-type: none"> REST permite la heterogeneidad mediante la intercomunicación debido a que muchos dispositivos programables aceptan peticiones HTTP
Interoperabilidad	<ul style="list-style-type: none"> Por medio de la utilización de estándares para la comunicación: XML, SOAP, WSDL, UDDI. El protocolo SOAP provee una estructura de mensajería para intercambiar mensajes XML entre plataformas heterogéneas. 	<ul style="list-style-type: none"> CORBA A través de la especificación de la interfaces mediante el lenguaje neutro IDL (Interface Definition language) las interfaces de los objetos se hace con independencia de la plataforma y de los lenguajes de implementación. 	<ul style="list-style-type: none"> Los contratos de los componentes se reducen a la definición de sus interfaces a nivel sintáctico, y la interoperabilidad se reduce a la comprobación de los nombres y perfiles de los métodos 	<ul style="list-style-type: none"> REST se compone de recursos, cada recurso es identificado mediante un Identificador único (URI), el cual puede ser accedido desde cualquier aplicación para ser manipulado
Reusabilidad	<ul style="list-style-type: none"> La reutilización de los servicios de negocio pueden ser expuestas como Servicios Web y ser reutilizadas para cubrir nuevas necesidades de negocio 	<ul style="list-style-type: none"> Puede ser cargado en tiempo de ejecución en un aplicación. 	<ul style="list-style-type: none"> Los componentes son diseñados para ser utilizados en escenarios diferentes por diferentes aplicaciones. COM permite la reutilización de objetos sin conocimiento de su implementación interna, porque fuerza a los implementadores de componentes a proveer interfaces bien definidas que están separados de la implementación 	<ul style="list-style-type: none"> Máxima reutilización porque todos los recursos tienen identificadores basados en una URI lo que la visibilidad de los recursos permite uso no provistos. Interconexión de recursos. Las representaciones de los recursos están interconectados mediante direcciones URL
Seguridad	<ul style="list-style-type: none"> Control de accesos integrado con estándares LDAP, SAML, y WS-Security. El uso de estándares SAML, WS-Security y XACML permiten autorización en base a reglas y puntos de control de políticas. 	<ul style="list-style-type: none"> Autenticación cliente/servidor generando certificados confirmando sus derechos. Auditoría de las invocaciones a métodos remotos por parte de los servidores 	<ul style="list-style-type: none"> El mecanismo de seguridad depende de la plataforma no de los componentes 	<ul style="list-style-type: none"> HTTPS se vale del protocolo criptográfico SSL el cual brinda autenticación de las partes y privacidad de la información transferida
Objetos de Aprendizaje / Ventajas	<ul style="list-style-type: none"> Permite integrar un repositorio en un sistema de búsqueda federada mediante la especificación SQL (Simple Query Interface) La utilización de los servicios web y las arquitecturas orientadas a servicios forman un parte fundamental de muchas de las normas que las principales instituciones publican dentro del mundo de e-learning Reutilización de servicios al disponibilizar los recursos a otros participantes en la red como servicios independientes a los que tienen acceso de un modo estandarizado. Constituye una tecnología muy adecuada para la implementación de repositorios que gestionan objetos de aprendizaje que se encuentran almacenados en diferentes dispositivos de recursos didácticos, ya que ofrece un acceso claro a los objetos distribuidos que están en los repositorios basados en diferentes tecnologías de almacenamiento de metadatos a través de una única interfaz 	<ul style="list-style-type: none"> Puede ser utilizado el broker como un middleware de comunicación de repositorios distribuidos. 	<ul style="list-style-type: none"> Puede ser utilizado como middleware dentro de una red de área local para interconectar diferentes repositorios de aprendizaje. 	<ul style="list-style-type: none"> Los objetos de aprendizaje son especificados como recursos únicos que son accedidos con una URL. Para agregar un nuevo recurso solo basta con definir el método PUT/DELETE
Objetos de Aprendizaje / Desventajas	<ul style="list-style-type: none"> Se debe soportar el protocolo SOA para poder interoperar los Objetos de aprendizaje 	<ul style="list-style-type: none"> No hay soporte HTTP para acceder a un recurso solo se soporta el protocolo TCP/IP por medio de interfaces IDL. Si se requiere comunicación con repositorios distribuidos muchas veces el uso de puertos generan problemas de conectividad por causa de puertos cerrados 	<ul style="list-style-type: none"> Para crear repositorios de aprendizaje con un middleware basado en componentes DCOM, se debe tener licencia propietaria Microsoft. Los Firewalls causan problemas con DCOM porque asigna un puerto por proceso y adicionalmente requiere que el puerto UDP y TCP se encuentre abierto. 	<ul style="list-style-type: none"> Para Brindar seguridad a los accesos de objetos de aprendizaje se deben generar certificados de seguridad para soportar el protocolo HTTPS

Tabla 2.1. Matriz comparativa de los diferentes estilos arquitectónicos

Partiendo de que un objeto de aprendizaje es un conjunto de recursos digitales, inherentemente una arquitectura orientada a recursos sería una solución ideal para implementar repositorios, pero actualmente no existe un acceso seguro, flexible y escalable.

La seguridad es uno de los aspectos más importantes que debe tener un repositorio distribuido, cualquier tipo de organización educativa o empresarial contempla un conjunto de normas mínimas que se deben implementar, mas allá de ofrecer un acceso flexible, la seguridad debe prevalecer y la arquitectura orientada a recursos tiene esta falencia respecto a la arquitectura orientada a servicios, si comparamos las primeras arquitecturas pioneras como fueron las arquitecturas orientadas a componentes (DCOM) y objetos (CORBA) que se basan en el protocolo RPC y las cuales se definen como arquitecturas totalmente seguras porque sus invocaciones están encapsuladas, presentan la gran desventaja como son las restricciones de los firewalls porque por cada proceso se asigna un puerto de comunicación, un problema que no se debe pasar por alto, además que son arquitecturas que tienen un alto nivel de acoplamiento porque se debe conocer el lenguaje sobre el cual están desarrolladas.

Si nos enfocamos en el beneficio, podemos destacar el rendimiento al procesar la información ya que la representación de los datos es binaria lo que no implica utilizar un transformador de mensajes, pero esto no es suficiente y por esta razón la arquitectura orientada a servicios se consolida como un candidato fuerte, SOA con el objetivo de reutilizar servicios ofrece un bajo acoplamiento, característica que es indispensable y que no me ofrecen las arquitecturas de componentes y objetos por sus interfaces IDL.

La interoperabilidad de plataformas heterogéneas es una ventaja necesaria porque reduce los costos de implementación, y SOA tiene en claro la explotación de la misma y junto con el estándar WS-Security provee las herramientas necesarias para garantizar la confidencialidad, integridad y la autenticación de mensajes.

Para implementar SOA se necesita el protocolo SOAP con el fin poder utilizar un WSDL que es un descriptor de servicio Web, o inclusive si utilizamos JMS (Java Message Service) para intercambiar los mensajes a través de una cola de peticiones, o inclusive si queremos adherir un nuevo repositorio hay que exponer una serie de servicios web donde la tarea no es fácil ya que se debe tener un conocimiento medio para implementarlos, y aquí es donde REST surge como candidato al ser una tecnología ligera que solo depende del protocolo http.

Con REST no se publican servicios RPC por lo que no se van a implementar interfaces con declaraciones de operaciones, lo que se publica son recursos, y estos recursos están representados en URIs, que es un mecanismo de identificación de recurso y que permite que sean accedidos fácilmente, el solo hecho que si queremos reemplazar un recurso, los clientes no tienen que realizar

cambios, ya que la URI del recurso sigue siendo la misma, este es un beneficio importante que nos brinda REST, y que sería un punto fuerte para decidir una arquitectura, pero con la matriz comparativa vemos que tomar una arquitectura como una unidad no es suficiente para cumplir con nuestro objetivo y es aquí donde surge el problema del presente trabajo.

3. DESCRIPCIÓN DEL PROBLEMA

Actualmente existe en la web una gran cantidad de objetos de aprendizaje para reutilizar y distribuir, la mayoría de estos recursos se encuentran almacenados en repositorios de aprendizaje independientes debido a que la interoperabilidad existente impide el crecimiento de las redes distribuidas y de las iniciativas para compartir los recursos.

Para adherir un repositorio en la actualidad, se requiere de un estilo Arquitectónico para lograr un acoplamiento con la red de repositorios distribuidos y lograr su funcionamiento. Este estilo, no satisface las características esenciales que debería tener un repositorio: escalable, flexible, seguro, de acceso abierto e integrado, debido a que la falencia técnica de un estilo se mejora con otro.

En cuanto a la búsqueda federada, se requieren mejoras y armonizaciones como son:

- Formato de intercambio de consultas
- Lenguaje de consulta abstracto e independiente de la estructura de los datos
- Descriptores de nivel de colección de contenidos de aprendizaje
- Mecanismos para resultados de la clasificación.

Por consiguiente las arquitecturas implementadas y disponibles no brindan un acceso flexible y seguro para las búsquedas distribuidas que se realizan para localizar información, en el caso de una determinada organización ya sea educativa o empresarial, para implementar un repositorio de aprendizaje distribuido existe una barrera en los detalles tecnológicos como la arquitectura soportada, la heterogeneidad y el alto acoplamiento de los componentes involucrados, por consiguiente surge la necesidad de transformar e innovar la manera como se interconectan los repositorios distribuidos y la manera como son accedidos para localizar información, con el fin de lograr una interoperabilidad limpia y transparente con un bajo grado de conocimiento para eliminar las limitaciones que impactan en el crecimiento tanto de las redes federadas existentes como las que están surgiendo.

Por lo que surgen los siguientes interrogantes para nuestro trabajo de investigación:

- Qué tipo de arquitectura brindaría un acceso flexible y seguro con un bajo nivel de acoplamiento entre sus componentes?
- Como convertir la adhesión de un objeto de aprendizaje en un proceso sencillo de implementar y que fomente el crecimiento de nuevas redes federadas?

- Como se deberían solucionar las barreras tecnológicas existentes para interoperar repositorios distribuidos en ambientes heterogéneos?
- Que tipo de implementación de seguridad deberían soportar los repositorios de aprendizaje para que el canal de compartición de recursos sea seguro ante ataques externos?

Profundizando aún más sobre la problemática abordada, surge la necesidad de proveer mecanismos tecnológicos que potencien y garanticen la interoperabilidad entre los repositorios de objetos de aprendizaje con el fin de posibilitar intercambios, migración y reutilización de objetos entre ambientes heterogéneos.

4. SOLUCION CONSIDERADA

En este capítulo se presenta la solución considerada (sección 4,1), y se discute su viabilidad (sección 4,2).

4.1. DESCRIPCION DE LA SOLUCION

La matriz analizada en la (sección 2.5), nos permite identificar que el problema radica en que las diferentes arquitecturas mencionadas, implementadas de forma independiente presentan distintas ventajas y desventajas lo cual da origen a la necesidad de potencializar la implementación con una propuesta de arquitectura híbrida (AHROA) que maximice los atributos de cada una de ellas y que provea como resultado una serie de beneficios que debería tener un repositorio de aprendizaje distribuido, lo que se pretende con el presente trabajo, además de lograr interoperar de una forma transparente cualquier tipo de arquitectura o plataforma heterogénea, bajo un mínimo nivel de acoplamiento, es que se generen iniciativas para que las redes federadas existentes se unan a una nueva red global federada sin restricciones de interoperabilidad o cierto grado de conocimiento para poder adherir un repositorio ya que esta es la principal causa que impacta en el crecimiento de las redes federadas existentes.

La arquitectura híbrida propuesta tiene las siguientes características:

- Abierta. Cualquier tipo de repositorio distribuido puede adherirse, con el objetivo de conformar una red federada de repositorios distribuidos.
- Escalable. Soporta el crecimiento de los repositorios adheridos sin afectar su disponibilidad.
- Integrada. Soporta cualquier tipo de integración bajo componentes o plataformas heterogéneas para poder interoperar.
- Flexible. Soporta cualquier tipo de cambios sobre los objetos de aprendizaje (Mecanismos de búsqueda, actualización de objetos, etc) dentro del contexto de los repositorios adheridos.
- Segura: Provee un canal seguro de comunicación debido a que implementa WS-Security para firmar los mensajes.

La arquitectura híbrida tiene como objetivo la interoperabilidad de repositorios de aprendizaje, por lo que estaría basada en el protocolo SOAP y el estilo REST, tomando como principio la convivencia de la arquitectura orientada a servicios y la arquitectura orientada a recursos,

consideraríamos que se garantizaría: interoperabilidad, escalabilidad, seguridad y rendimiento, en la figura 4.1 se observa los componentes que conforman la arquitectura híbrida

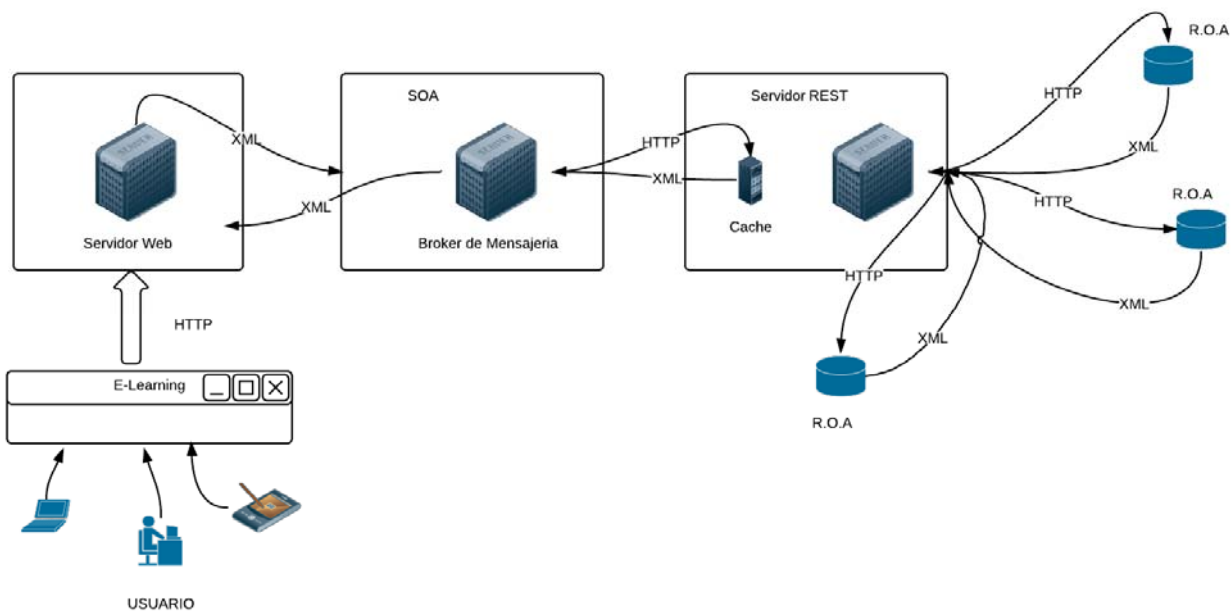


Figura 4.1 Arquitectura Híbrida (AHROA)

Los componentes involucrados en la arquitectura híbrida son:

- Servidor web accedido a través de interfaz de usuario: Es el canal de comunicación con el usuario al realizar la solicitud de búsqueda y con el Bróker de mensajería para enviar la solicitud del mismo.
- Bróker de mensajería: es el encargado de transformar y realizar la solicitud al servidor REST
- Servidor REST: Es el encargado de derivar la solicitud al cache indexador para que valide si la URI se encuentra almacenada, en caso de no encontrarse en el cache, buscara en cada repositorio disponible.
- Cache Indexador de Peticiones HTTP: Es el encargado de verificar si la URI ya ha sido almacenada con anterioridad ej: <http://www.roat.com/AHROA/repository?find=campo&...>
- Repositorios de Objetos Aprendizaje: Son las Bases de datos donde se encuentran los objetos de aprendizaje

El usuario por medio de una interface web realiza la consulta del objeto de aprendizaje que desea ver, esta comunicación se realiza a través del protocolo HTTP, inmediatamente el servidor web se conecta al bróker de mensajería para enviar la solicitud del usuario en formato XML esta fase se conoce como serialización que es el proceso de convertir la información del estado de un objeto en

formato XML , el bróker de mensajería realiza la transformación del mensaje de XML a Binario para poder realizar una solicitud al Servidor REST, esta solicitud se deriva primero al cache indexador que se encarga de buscar la URI para validar si la consulta se realizó previamente, si la URI no se encuentra almacenada, se realiza una petición a la lista de repositorios de aprendizaje disponibles, esta consulta se realizara secuencialmente, repositorio por repositorio, cuando el criterio de búsqueda es encontrado se devolverá la respuesta en formato XML, el servidor REST tomara la respuesta XML para devolvérsela al Bróker de Mensajería, donde el Bróker devolverá la respuesta XML al servidor Web que inicialmente realizo la petición, en este paso el servidor Web aplica una deserialización para mostrar la respuesta en la interfaz del usuario.

La interrelación de los componentes basa su objetivo en la convivencia de la arquitectura orientada a servicios y la arquitectura orientada a recursos, con esta solución se estaría garantizando:

- Interoperabilidad: por medio de una búsqueda unificada se está obteniendo la información dentro de múltiples repositorios distribuidos.
- Escalabilidad: la adhesión de un nuevo repositorio a la red no requiere crear servicios web complejos solo basarse en el formato HTTP bajo el estilo REST.
- Seguridad: Se están aplicando todos los beneficios de SOA por medio del estándar WS-Security
- lo que me garantiza que no se modificaran los mensajes de respuesta y/o envío.
- Rendimiento: El cache indexador bajaría la tasa de solicitudes de búsqueda, y el estilo
- REST garantiza que la mensajería sea liviana.

4.2. DISCUSION SOBRE LA VIABILIDAD LA SOLUCION

Los repositorios distribuidos son de gran importancia, por la gran cantidad de información que existe en la web y que se necesita compartir para evitar la duplicidad de contenidos, a tal grado que se promueva el principio de la reutilización.

La interoperabilidad es el factor más importante dentro del contexto de repositorios distribuidos, la limitación de interoperabilidad hace que no sea posible una solución distribuida y que muchas veces se centralice la información, por lo que los costos aumentan sobre todo en el mantenimiento de los mismos, solo con pensar el ámbito geográfico, las búsquedas de contenido, dependerían de la información de que ha sido almacenada en un solo repositorio, limitando el universo geográfico de información, la interoperabilidad de repositorios distribuidos es una necesidad actual dentro del

ámbito educativo y empresarial, por lo tanto, considero que con la implementación de la solución propuesta, estamos maximizando todos los beneficios que nos brinda la Arquitectura orientada a servicios y recursos, además, la arquitectura propuesta, es sencilla de implementar, y su mayor ventaja, es que es muy fácil adicionar nuevos repositorios, las redes federadas de repositorios de aprendizaje se basan en un diseño preliminar de servicios, por lo que agregar un nuevo repositorio es un proceso complejo, con esta arquitectura estamos evolucionando la manera de interoperar repositorios distribuidos.

En términos de seguridad como beneficio, se ofrece una canal de comunicación seguro, al utilizar servicios web, a través del protocolo SOAP junto con el estándar JAX-WS security, el cual nos ofrece: autenticidad, integridad y confidencialidad de los mensajes, con el fin de evitar ataques externos y violación de la confidencialidad de información dentro del contexto empresarial, refiriéndonos a una solución privada, necesidad que hoy en día no se tiene en cuenta, a la hora de implementar por falta de conocimientos técnicos.

Esta arquitectura ofrecería una búsqueda integrada de localización de información como el mayor beneficio, gracias a la flexibilidad con la que se puede adherir un nuevo repositorio, terminando con la limitante de difíciles implementaciones o dependencias en la adhesión de repositorios, es una arquitectura adaptable y accesible al ser cliente/servidor y utilizar REST en la búsqueda de información en los repositorios distribuidos, con bajo grado de acoplamiento al implementar servicios web, lo que la viabilidad de la solución está fundamentada en su simplicidad de implementación, lo que generaría iniciativas de redes federadas y conectores de integración para los ambientes virtuales de aprendizaje.

5. CONCLUSIONES

En este trabajo de especialidad, se presentaron las características que formarían parte de la arquitectura híbrida que nos permitirá interoperar repositorios distribuidos de objetos de aprendizaje en ambientes heterogéneos de una forma flexible y segura.

Se realizó una matriz comparativa la cual fue el punto de partida para analizar el estado del arte para generar una discusión sobre las ventajas y desventajas existentes de una implementación en forma individual y plantear una posible solución.

Partiendo del objetivo principal del trabajo, se ha logrado como resultado, la propuesta de una arquitectura híbrida que implementa la arquitectura orientada a servicios (SOA) y la arquitectura orientada a recursos, generando como beneficio que la interoperabilidad sea realizada en un canal de comunicación seguro gracias a los estándares de seguridad que ofrece SOA.

La arquitectura orientada a recursos por medio del estilo Rest nos ofrece desempeño, permitiendo integrar de forma simple un repositorio distribuido mediante una URI o identificación de recurso, todo esto por medio de una simple petición HTTP utilizando el método GET como consulta, lo que reduce la complejidad de la integración técnica.

La Arquitectura propuesta resuelve el problema interoperabilidad de repositorios distribuidos en ambientes heterogéneos bajo un alto nivel de desacoplamiento de componentes de forma efectiva y eficiente, donde se genera valor en la reutilización de objetos de aprendizaje mediante su publicación y localización universal para que las redes federadas existentes, se unan a esta nueva arquitectura, para generar una nueva red, donde la adhesión de nuevos repositorios se realice de forma flexible y sin depender de ciertas condiciones técnicas para su implementación.

Las características más importantes de la arquitectura propuesta son:

- Es una arquitectura abierta que se basa en estándares y protocolos para poder interoperar e integrar repositorios distribuidos de objetos de aprendizaje.
- Ofrece un canal de comunicación seguro ante ataques o eventos externos que permitan indisponibilizar el servicio.
- La adhesión de un nuevo repositorio se realiza de una forma flexible y ágil, al no depender de un acoplamiento entre los componentes arquitecturales involucrados.
- Fomenta la reutilización de los objetos de aprendizaje.

- Se basa en la publicación y localización de cualquier objeto de aprendizaje independiente de la arquitectura o ambiente donde se encuentra almacenado.

Se espera que en un próximo trabajo, se generen alternativas para mejorar la localización de objetos de aprendizaje en repositorios distribuidos, mediante búsquedas semánticas en la arquitectura híbrida propuesta.

6. REFERENCIAS

- Mazo, S., Otón, S., De-Marcos, L., García, A., García, E., (2012). *Restful Service Oriented Architecture for querying and publishing Learning Objects in Repositories*. ELML 2012: The Fourth International Conference on Mobile, Hybrid, and online Learning.
- Montilva, J., Rojas, M., Orjuela, A., (2011). *RDOA-WS: repositorio distribuido de objetos de aprendizaje soportado con servicios web*. Revista avances en sistemas e informática. Vol. 8: 183-189
- Motz, R., Saavedra, R., Vallespir, D., (2010). *Localizador de Objetos de Aprendizaje Distribuidos*. Congreso Iberoamericano de Informática Educativa.
- Oton, Salvador., Ortiz, A., Hilera, J., (2007). *SROA: Sistema de reutilización de objetos de aprendizaje*. ISSN: 1699-4574. Revista Iberoamericana de Informática Educativa. Vol. 5: 7-22
- Solarte, M., Martinez, J., (2003). *Enfoque Arquitectónico basado en componentes para la construcción de Ambientes Virtuales de Aprendizaje*. Tercer Congreso Iberoamericano de Telemática.
- Tabares, V., Rodríguez, P., Duque, N., Moreno, J., (2012). *Modelo Integral de Federación de Objetos de Aprendizaje en Colombia - más que búsquedas centralizadas*. Séptima Conferencia Latinoamericana de Objetos y Tecnologías de Aprendizaje. Pág. 410-418
- Trujillo, J., Espinoza, A., (2010). *Conceptos fundamentales de Ingeniería dirigida por Modelos y Modelos de Dominio Específico*. Revista de Investigación de Sistemas e Informática. Vol. 7: 9-19. ISSN 1815-0268

