

Universidad Central de Venezuela
Facultad de Ciencias
Escuela de Computación



***Integración del Repositorio de
AMBAR con el Repositorio de
Metadata a través de la Capa de
Servicios***

Trabajo especial de Grado presentado ante la ilustre
Universidad Central de Venezuela por la Bachiller:

Quintero Sanella, Joseline Adriana

Para optar al título de Licenciado en Computación

Tutor(es):

Prof. Yosly Hernández B.

Prof. Nora Montaña

Caracas, 30 de Abril de 2009

ACTA

Quienes suscriben, miembros del Jurado designado por el Consejo de Escuela de Computación, para examinar el Trabajo Especial de Grado presentado por la Bachiller Joseline Adriana Quintero Sanella C.I. 16.083.076, con el título: “Integración del Repositorio de AMBAR con el Repositorio de Metadata a través de la Capa de Servicios”, a los fines de optar al título de Licenciado en Computación, dejen constancia de lo siguiente:

Leído como fue, dicho trabajo por cada uno de los miembros del jurado, se fijó el día 30 de Abril de 2009 a la 09:00am, para que sus autores lo defendieran en forma pública, lo que hicieron en el Sala 1 de la Escuela de Computación, mediante una presentación oral de su contenido, luego de lo cual respondieron las preguntas formuladas. Finalizadas la defensa pública del Trabajo Especial de Grado, el jurado decidió aprobarlo.

En fé de lo cual se levanta la presente Acta, en Caracas los 30 días del mes de abril del año dos mil nueve dejándose también constancia de que actuó como Coordinador del Jurado la Profesora tutor Yosly Hernández.

Prof. Yosly Hernández (Tutor)

Prof. Nora Montaña (Tutor)

Prof. Claudia León (Jurado)

Prof. Doris Pernalette (Jurado)

Prof. Vanessa Miguel (Jurado Suplente)

Prof. Antonio Silva (Jurado Suplente)

RESUMEN

El presente trabajo consiste en la integración de la base de datos (BD) del SIG-AMBAR con el Repositorio de Metadata a través de la Capa de Servicios Web, asociada sólo a los Objetos de Aprendizaje.

Actualmente, AMBAR es un proyecto que posee dos aplicaciones, donde cada aplicación maneja información sobre el mismo recurso, el cual es llamado Objeto de Aprendizaje (OA). Por un lado, se tiene la base de datos (BD) del SIG-AMBAR que almacena OA, pero no posee una organización estructurada de los atributos que describen a estos, es decir, los metadatos. Y por otro lado, se tiene el repositorio de Metadata el cual posee toda una estructura estandarizada de los atributos que describen a los OA, por tanto, al integrar estas aplicaciones, se está facilitando la búsqueda, recuperación, catalogación y administración de los OA.

La definición de los Servicios Web para el acceso a la BD del SIG-AMBAR mejora la interoperabilidad entre las diferentes aplicaciones que constituirán el sistema, ya que éstos ofrecen una alternativa de software independiente en cuanto a la plataforma, basada en estándares para la integración de aplicaciones, la automatización de procesos de negocio y la publicación de la información de diversas fuentes. El contar con una infraestructura integrada, segura, escalable y disponible, disminuye los costos y permite compartir información de manera confiable.

La integración de los repositorios de metadata y de OA, proporciona un espacio para manipular OA que pueden ser almacenados y recuperados de manera eficiente. Los objetos ahora serán descritos a través de una metadata que obedece a las especificaciones de un estándar mundial especializado para describirla, llamado Learning Object Metadata (LOM), que permite identificarlos para luego potenciar su búsqueda, evaluación, obtención y reutilización.

Las tecnologías y herramientas utilizadas por ambas aplicaciones contribuyeron al desarrollo exitoso de la integración. La tecnología JSP, que permite generar contenido dinámico para ambiente Web con programación en java, lenguaje orientado a objetos para el desarrollo de los Servicios Web; Servidor de Apache Tomcat, servidor que se

utilizó como contenedor Web de la aplicación; Axis y Axis2, implementación de SOAP (Simple Object Access Protocol) que proporciona el entorno para la ejecución y publicación de los Servicios Web, entre otros.

Es importante resaltar que el proceso de integración se realizara únicamente sobre los Objetos de Aprendizaje debido a que el repositorio de metadata sólo almacena información descriptiva sobre ellos y no información adicional.

Palabras Claves: Objetos de aprendizaje (OA), Metadata, Repositorio de Objetos de Aprendizaje, Repositorio de Metadata, Servicios Web, LOM.

ÍNDICE GENERAL

ÍNDICE DE FIGURAS	7
ÍNDICE DE TABLAS	8
INTRODUCCIÓN.....	9
CAPÍTULO 1. PROBLEMA DE INVESTIGACIÓN	11
1.1. Problema de Investigación	11
1.2. Objetivo General.....	11
1.3. Justificación	12
1.4. Preguntas de Investigación	12
1.5. Objetivos Específicos	13
1.6. Antecedentes.....	13
1.7. Método de Desarrollo de Software	14
CAPÍTULO 2. MARCO CONCEPTUAL.....	16
2.1. Objetos de Aprendizaje	16
2.1.1. <i>Características de los Objetos de Aprendizaje</i>	17
2.1.2. <i>Funciones</i>	17
2.1.3. <i>Taxonomía general de los Objetos de Aprendizaje</i>	18
2.1.4. <i>Ventajas de los Objetos de Aprendizaje</i>	19
2.2. Repositorios de Objetos de Aprendizaje.....	20
2.2.1. <i>Tipos de ROA</i>	22
2.2.2. <i>Características de los Repositorios de Objetos de Aprendizaje</i>	24
2.2.3. <i>Funciones de los Repositorios</i>	25
2.2.4. <i>Ejemplos de Repositorios de Objetos de Aprendizaje</i>	25
2.3. Estándares para Objetos de Aprendizaje	26
2.3.1. <i>SHARABLE CONTENT OBJECT REFERENCE MODEL (SCORM)</i>	26
2.3.2. <i>LEARNING OBJECT METADATA (LOM)</i>	27
2.4. Proyecto AMBAR.....	30
2.4.1. <i>Visión Interdisciplinaria de AMBAR</i>	30
2.4.2. <i>Características de AMBAR</i>	31

CAPÍTULO 3. ANÁLISIS SOBRE LA BASE DE DATOS Y EL REPOSITORIO DE METADATA DE AMBAR	36
1. <i>Base de Datos de AMBAR</i>	36
2. <i>Repositorio de Metadata</i>	41
CAPÍTULO 4. MARCO METODOLÓGICO	46
4.1. Primera Iteración	48
<i>Fase de Inicio</i>	49
<i>Fase de Integración</i>	49
<i>Fase de Transición</i>	66
4.2. Segunda Iteración	66
<i>Fase de Inicio</i>	66
<i>Fase de Integración</i>	67
<i>Fase de Transición</i>	82
CONCLUSIONES.....	83
RECOMENDACIONES	84
REFERENCIAS BIBLIOGRÁFICAS	85
ANEXOS.....	89
Manual del desarrollador e instalación del Repositorio de Metadata	89
Manual del desarrollador e instalación de la Capa de Servicios de AMBAR	101

ÍNDICE DE FIGURAS

Figura 1. Interacción de los ROA.....	222
Figura 2. Tipos de ROA por la distribución de los OA	22
Figura 3. Tipos de ROA por la distribución de los metadatos.	23
Figura 4. Estructura Interdisciplinaria de AMBAR.	31
Figura 5. Diagrama de Clases a nivel macro del repositorio de OA.....	33
Figura 6. Diagrama de Casos de Uso (funcionalidades principales y sus usuarios potenciales) de AMBAR.....	34
Figura 7. Modelo conceptual de la BD de AMBAR.....	36
Figura 8. Diagrama conceptual del repositorio de Metadata	42
Figura 9. Arquitectura de la integración de la BD del SIG-AMBAR con el repositorio de Metadata.....	61
Figura 10. Creación de un Servicio Web	64
Figura 11. Ingreso de nombre y ubicación del Servicio Web	64
Figura 12. Modelo Conceptual de la integración	69
Figura 13. Pantalla inicial - Capa de Servicios	75
Figura 14. Autenticación Usuario Administrador	76
Figura 15. Página principal Administrador.....	77
Figura 16. Página Administrador – Otros Servicios Web	78
Figura 17. Agregar OA y su información descriptiva	78
Figura 18. Agregar OA – Operación exitosa.....	79
Figura 19. Búsqueda de OA – Modificar.....	79
Figura 20. Resultado de búsqueda de OA a modificar.....	80
Figura 21. Modificar OA.....	81
Figura 22. Ver recurso digital	84

ÍNDICE DE TABLAS

Tabla 1. Caso de Uso “Recorrer el Sitio Web”	53
Tabla 2. Caso de Uso “Registrar Usuario”	54
Tabla 3. Caso de Uso “Autenticar Usuario”	55
Tabla 4. Caso de Uso “Manipular Elemento”	56
Tabla 5. Caso de Uso “Almacenar OA”	57
Tabla 6. Caso de Uso “Mostrar OA”	58
Tabla 7. Caso de Uso “Modificar OA”	58
Tabla 8. Caso de Uso “Eliminar OA”	60

INTRODUCCIÓN

La cantidad de información disponible en Internet ha ido en aumento y las tecnologías van facilitando nuevos recursos para la gestión de la información, los servicios adicionales que pueden ofrecerse para el uso de los recursos van más allá de la simple consulta y organización mediante catálogos.

La gestión de información ha sido una actividad que ha permitido a través de los años que datos y contenidos sean utilizados para diversas aplicaciones, por diversos grupos e incluso en diferentes países alrededor del mundo.

Con la aparición de los Objetos de Aprendizaje (OA) surgieron necesidades particulares para gestionar los contenidos educativos, y los Repositorios de OA comienzan a posicionarse como importantes herramientas que tienen como función resguardar los recursos, hacerlos disponibles para diversos usos y para compartirlos con otras aplicaciones, facilitando con esto el flujo de contenidos y la expansión de servicios.

El objetivo de este trabajo es la integración y adaptación del Repositorio de AMBAR y de metadata a través de la capa de Servicios Web de base de datos, para así permitir en ella, la incorporación y actualización de metadata bajo el estándar LOM implementado en el repositorio de Metadata.

En este trabajo de investigación existen 3 capítulos, los cuales se describen a continuación:

El primer capítulo, describe cuál es el problema de investigación en el que se enmarca este trabajo, el objetivo general y objetivos específicos, además de la justificación de la investigación.

El segundo capítulo, comprende el marco conceptual de la investigación, donde se describen todos los aspectos relacionados con las aplicaciones que se integraron. Entre ellos, están los Objetos de Aprendizaje (OA), Repositorios de OA y los estándares para OA, tales como: ADL SCORM, Learning Object Metadata (LOM).

En el tercer capítulo, se describe de manera detallada cada uno de los pasos del método de desarrollo Agile Unified Process (UP Ágil). Este método consta de cuatro fases: Inicio, Elaboración, Construcción y Transición, para el proceso de integración.

Por último, se presentan las conclusiones del trabajo y algunas recomendaciones.

CAPÍTULO 1. PROBLEMA DE INVESTIGACIÓN

1.1. Problema de Investigación

Actualmente, existe el Sistema Generador de AMBientes de Enseñanza-ApRendizaje Constructivistas basados en Objetos de Aprendizaje (SIG-AMBAR) el cual esta explicado en el Capítulo 2. Pág. 29, y este cuenta con una capa de servicios para la base de datos (BD) definida en DB4Objects. El modelo conceptual correspondiente a la BD, define la estructura del repositorio del SIG-AMBAR, en el cual se encuentra definida, entre otros elementos, metadata que se va a almacenar de Objetos de Aprendizaje (OA). Ahora bien, esta metadata que se almacena corresponde a cierta información descriptiva no estandarizada, la cual no es suficiente para describir y catalogar al OA, y por consiguiente, dificulta las búsquedas y recuperación del mismo.

Por otro lado, existe un Repositorio de Metadata, independiente de la BD descrita anteriormente, que posee toda una estructura estandarizada de definición, descripción y almacenamiento de los recursos, que a su vez, proporciona mecanismos eficientes de localización, catalogación y recuperación de los OA.

Por tanto, ¿Cómo realizar la integración del Repositorio de OA y el repositorio de metadata a través de los servicios web existentes, que permita aprovechar los beneficios que el repositorio de metadata posee?

1.2. Objetivo General

Integrar el Repositorio de Metadata con el Repositorio de OA a través de la Capa de Servicios Web, para lograr que los OA posean mecanismos eficientes para su catalogación, descripción y localización.

1.3. Justificación

Este proceso de integración es necesario, debido a que en el SIG-AMBAR existen dos aplicaciones que manipulan OA de forma independiente sin relación entre sí, desaprovechando las bondades que una aplicación le pudiera brindar a la otra. Una de las aplicaciones, es la base de datos del SIG-AMBAR (Accedida desde la Capa de Servicios), la cual es capaz de almacenar y gestionar, tanto los OA como los ambientes de aprendizaje (que representan los andamiajes que son capaces de desarrollar múltiples competencias en los aprendices que lo van a utilizar). La otra aplicación, es el Repositorio de Metadata, el cual es capaz de almacenar y manejar metadata de OA a través de una estructura estandarizada. Ver Capítulo 3 para mayor información sobre las dos aplicaciones antes descritas.

Este trabajo se desarrollará en el marco del proyecto de investigación que se lleva a cabo en el laboratorio de Bases de Datos de la Facultad de Ciencias de la Universidad Central de Venezuela, llamado “Sistema Generador de AMBientes de Enseñanza-ApRendizaje Constructivistas basados en Objetos de Aprendizaje (SIG-AMBAR)”. El resultado de la integración permitirá que los OA, manejados a través de la Capa de Servicios de la BD del SIG-AMBAR, estén catalogados y descritos siguiendo un estándar certificado mundialmente llamado Learning Object Metadata (LOM) y a su vez, estos objetos, puedan ser accedidos a través de una aplicación Web que provea mecanismos eficientes para la manipulación y localización de los mismos.

Ahora bien, los OA deben estar almacenados en repositorios y etiquetados con su respectiva metadata ya que los objetos aislados no tienen ninguna relevancia ni significado real ya que no podrían ser reutilizados. De esta forma, integrando el Repositorio de Metadata con la BD del SIG-AMBAR a través de la Capa de Servicios Web se fortalece aun más el sistema, manteniendo así, el uso de estándares y facilidades mundialmente establecidas en el proceso de enseñanza y aprendizaje por internet.

1.4. Preguntas de Investigación

- ¿Qué elementos deben ser modificados en la Capa de Servicios de la Base de Datos de AMBAR?

- ¿Cómo se manejará el control de acceso al Repositorio de Metadata al integrarlo con la base de datos de AMBAR?
- ¿Qué adaptaciones se deben realizar en el Repositorio de Metadata para que permita la integración?
- ¿Cómo será la interfaz de usuario definitiva?

1.5. Objetivos Específicos

- Adaptar el modelo conceptual de la base de datos de AMBAR para que permita la integración con el Repositorio de Metadata.
- Implementar un control de acceso único a los dos repositorios.
- Evaluar interfaces de las aplicaciones que permitan acceder a los repositorios.
- Configurar los servicios de los repositorios en función de la integración a realizar.

1.6. Antecedentes

Actualmente hay muchas investigaciones referentes a los OA. Algunos de estos estudios destacables que siguen el paradigma de SMBDOO que se relacionan directamente con la integración que se implementa, son los siguientes:

- Beleño, Cira (2007). "Capa de servicios de base de datos del Repositorio del Sistema Generador de AMBientes de Enseñanza-ApRendizaje AMBAR". Caracas: Universidad Central de Venezuela. Donde se analizó, diseñó e implementó la Capa de Servicios de la base de datos de AMBAR, logrando que pueda tener un bajo acoplamiento con los demás componentes de su arquitectura. Es un nuevo sistema interoperable entre sus aplicaciones de software, que proporciona independencia entre sus módulos, además las diferentes aplicaciones que constituirán el sistema podrán hacer uso de la base de datos sin preocuparse por los lenguajes de programación utilizados en su desarrollo, ubicación o plataforma en que fueron implementadas.
- Nieves, C., & Alegría, R. (2008). "Repositorio de metadata de los objetos de aprendizaje del Sistema Generador de AMBientes de Enseñanza-ApRendizaje Constructivistas basados en Objetos de Aprendizaje". Caracas: Universidad

Central de Venezuela. Donde el Repositorio de Metadata es basado en el estándar LOM y es accedido a través de una aplicación Web que provee mecanismos eficientes para la descripción y localización de los objetos de aprendizaje; dicha aplicación se desarrolló mediante el desarrollo y el despliegue de Servicios Web, para ofrecer una capa de servicios de interconexión con otros sistemas.

1.7. Método de Desarrollo de Software

Los métodos asignan un proceso disciplinado sobre el desarrollo de software con el fin de hacerlo más previsible y eficiente. Como consecuencia de estos métodos surgió un nuevo grupo, adoptando el término “métodos ágiles” que representa su característica primordial, adaptabilidad para dar respuesta pronta y eficaz a los inevitables cambios.

Para el desarrollo del Trabajo Especial de Grado se utilizará el método Proceso Unificado Ágil (UP Ágil). Este método es una versión simplificada del Proceso Unificado de Rational (RUP), en el cual se describe de una manera simple el desarrollo de software de aplicaciones de negocio utilizando técnicas y conceptos ágiles.

UP Ágil divide el proceso de desarrollo en 4 fases:

1. Inicio: El objetivo de esta fase es obtener una comprensión común cliente-equipo de desarrollo del alcance del nuevo sistema y definir una o varias arquitecturas candidatas para el mismo. Es decir, es una visión aproximada, estudio económico, alcance y tiempos.
2. Elaboración: El objetivo es que el equipo de desarrollo profundice en la comprensión de los requisitos del sistema y en validar la arquitectura. Es decir, es una visión refinada, permite la implementación iterativa del núcleo de la arquitectura, solución de los riesgos mayores, identificación de la mayoría de requerimientos y del alcance, estimaciones más realistas.
3. Construcción: Durante la fase de construcción el sistema es desarrollado y probado al completo en el ambiente de desarrollo. Es decir, Implementación iterativa de los elementos que quedan, más fáciles y con menor riesgo.

4. Pruebas: el sistema se lleva a los entornos de preproducción donde se somete a pruebas de validación y aceptación y finalmente se despliega en los sistemas de producción. Es decir, pruebas beta, entrega de producto, adiestramiento de usuarios.

Una de las razones por las cuales se utilizó UP Ágil, es que plantea el desarrollo de prototipos ejecutables durante la base de elaboración del producto, donde se demuestre la validez de la arquitectura para los requisitos clave del producto y que determinan los riesgos técnicos.

CAPÍTULO 2. MARCO CONCEPTUAL

Los OA son un nuevo concepto de creación de recursos digitales educativos los cuales se basan en estándares y se almacenan en repositorios para su reutilización y localización.

En este capítulo, se describirán los conceptos necesarios de los elementos utilizados para la realización del proceso de integración de los repositorios en AMBAR. Dentro de los elementos utilizados se encuentra, como base de todo el proyecto, los OA, como éstos pueden ser almacenados y que estándares son utilizados para la manipulación de los mismos.

2.1. Objetos de Aprendizaje

El Comité de Estándares en Tecnología de Aprendizaje de la IEEE define un OA como: “Una entidad digital o no digital, la cual puede ser usada, reusada o referenciada durante el aprendizaje basado en la tecnología.”

Por otra parte, APROA(2005) define un OA como: “La mínima estructura independiente que contiene un objetivo, una actividad de aprendizaje, metadatos y un mecanismo de evaluación, la cual puede ser desarrollada con tecnologías de infocomunicación (TIC) con el fin de posibilitar su reutilización, interoperabilidad, accesibilidad y duración en el tiempo”.

Wiley define OA como: “Elementos de un nuevo tipo de instrucción basada en el computador y fundamentada en el paradigma computacional de la “orientación a objeto” (Wiley, 2000). Se valora sobre todo la creación de componentes (llamados “objetos”) que pueden ser reutilizados en múltiples contextos. Esta es la idea fundamental que se esconde tras los OA: los diseñadores instruccionales pueden construir pequeños componentes de instrucción (en relación con el tamaño de un curso entero) que pueden ser reutilizados varias veces en contextos de estudio diferentes.

Este trabajo de investigación se basa en el concepto descrito anteriormente de Wiley (2000).

2.1.1. Características de los Objetos de Aprendizaje

Los OA poseen un conjunto de características clave. A continuación se presentan las que describen García y Galeana (2004):

- Capacidad de ser reutilizado.
- Capacidad de generar aprendizaje.
- Proporcionar interoperabilidad.
- Facilidad de acceso a contenidos apropiados en tiempos apropiados.
- Vigencia de la información de los objetos, sin necesidad de nuevos diseños.
- Capacidad de construir contenidos y objetos nuevos, derivados de él.
- Capacidad de ser actualizados o modificados, aumentando sus potencialidades a través de la colaboración.
- Flexibilidad, versatilidad y funcionalidad, con elasticidad para combinarse en diversas propuestas de áreas del saber diferentes.
- Interactividad, permitiendo generar actividades y comunicación entre los sujetos involucrados.
- Acoplamiento a las necesidades de aprendizaje de cada individuo.
- Autocontención conceptual, capacidad de auto explicarse y posibilitar experiencias de aprendizaje integral.
- Escalabilidad, permitiendo la integración con estructuras más complejas.

Las características de los OA se heredan, esto evita que los profesores vuelvan a crear recursos de aprendizaje que ya existen o desde cero y, además, les da la oportunidad de distribuir lo que ellos generan.

2.1.2. Funciones

Según Galeana (2004), las funciones que debe tener un OA son:

- Favorecer la generación, integración y reutilización de un OA.

- Estimular el estudio autogestivo.
- Promover el trabajo colaborativo.
- Posibilitar el acceso remoto a la información y contenidos de aprendizaje.
- Posibilitar la integración de diferentes elementos multimedia a través de una interfaz gráfica.
- Contribuir a la actualización permanente de profesores y alumnos.
- Estructuración de la información en formato hipertextual.
- Facilitar la interacción de diferentes niveles de usuarios (administrador, diseñador, alumno).

Las funciones en general deben satisfacer las expectativas de todos los usuarios que referencian, crean o reutilizan los OA, es decir que la funcionalidad de un OA dependerá de para que el usuario lo necesite. Pero la función principal de un OA es su reusabilidad, para así fomentar la expansión del conocimiento y de la cooperación a nivel educacional que estos pueden brindar, y evitar la creación de un objeto que ya fue creado con anterioridad.

2.1.3. Taxonomía general de los Objetos de Aprendizaje

Existen diferentes clasificaciones de los OA, dentro de las más importantes se puede mencionar la taxonomía de Wiley (2000), que plantea tipos de OA:

- **Fundamentales:**
Recursos digitales individuales sin combinar con cualquier otro. Es generalmente una ayuda visual que permite ejemplificar o mostrar la información. Ejemplo de ellos Video, Audio, Imagen, Texto, etc.
- **Combinados Cerrados:**
Número pequeño de recursos digitales combinados, cuyos OA constituyentes no son accesibles individualmente para la reutilización de este OA. Este tipo de OA puede contener lógica limitada, pero no debe contener la lógica interna compleja puesto que esta capacidad no sería reutilizable en otros OA. Ejemplo: Un video con Audio, Una imagen con Texto, etc.

- **Combinados Abiertos:**
Gran número de recursos digitales combinados, cuyos componentes pueden ser directamente accedidos para su reutilización. Una página Web es un ejemplo, sus imágenes, videos, y otros medios existen en un formato reutilizable y se combinan en un OA a través de los requerimientos en el tiempo.
- **Presentación Generativa:**
Es la estructura lógica para combinar, generar o reorganizar los objetos de aprendizaje de nivel inferior. Los objetos deben ser capaces de pasar y recibir mensajes de otros objetos. Estos objetos tienen un alto grado de reusabilidad intra-Contextual, por ejemplo describir el número de veces que objeto de aprendizaje puede ser re usado en el mismo contexto o dominio.
- **Generativos Instruccionales:**
Proporcionan la estructura lógica para combinar las diferentes taxonomías de los OA y evaluar las interacciones del estudiante con estas combinaciones, las cuales son creadas para apoyar la instanciación de las estrategias educacionales, como por ejemplo “recuerde y realice una serie de pasos”.

2.1.4. Ventajas de los Objetos de Aprendizaje

Dentro de las ventajas más relevantes de los OA, se tiene:

- Son reutilizables. Deben poder ser utilizados en contextos diferentes y aun mantener su significado.
- Son acoplables. Debe permitir conectarse con otros objetos de aprendizaje para definir un objeto más complejo.
- Granularidad. Los OA son pequeñas unidades de información cuyos componentes están fuertemente cohesionados.
- Independencia de otros objetos. Deben cumplir con el objetivo de su diseño sin requerir de otros objetos externos.
- Capaces de soportar cambios tecnológicos. Deben ser independientes de la tecnología empleada sin necesidad de ser rediseñados una vez se produzcan cambios tecnológicos.

- Interoperables. Proporcionan la flexibilidad de utilizar componentes desarrollados con otras plataformas y otras herramientas.
- Personalización. El diseño de materiales formativos en forma de OA, facilita la personalización del contenido al permitir la recombinación de materiales a la medida de las necesidades formativas del colectivo o de los individuos concretos a quienes van dirigidos. De esta forma la adaptación a las necesidades formativas de cada alumno es absoluta permitiendo ofrecer, a cada uno, los contenidos formativos que precisa en cada momento.
- Flexibilidad. Un material diseñado para ser usado en múltiples contextos puede ser reutilizado con mucha más facilidad que un material que ha de ser reelaborado para cada nuevo contexto.

Un docente puede adoptar la estrategia de aceptar nuevos retos laborales, como puede ser la incorporación de OA basados en la Web, con el fin de enriquecer su experiencia laboral y así crecer como profesional. ¿Cuáles son, en términos generales, los potenciales beneficios de trabajar con OA? Varas (2002) promueve la utilización de OA "para evitar redescubrir soluciones descubiertas antes; para aumentar el alcance y los beneficios de cada buena solución; porque se pueden construir materiales de primera calidad con menos esfuerzo; porque los derechos de uso (copyright) son claros; porque son fáciles de acceder y compartir."

El hecho de que los OA se presenten como unidades mínimas modulares hace que el docente pueda elegir aquellos que se adapten fácilmente a su estilo de enseñanza, a la diversidad de estilos de aprendizaje de sus alumnos y a su programa de estudios.

2.2. Repositorios de Objetos de Aprendizaje

En el estudio sobre OA se ha insistido en que la reutilización debe ser una de sus principales bondades, es entonces cuando se habla de tener un lugar destinado para su almacenamiento y clasificación para facilitar posteriormente su mantenimiento, localización y, posiblemente, también compartir ese OA con otros sistemas para aplicaciones diversas. Este almacén es lo que se conoce como Repositorio de OA (ROA). Los sistemas de repositorios son la infraestructura clave para el desarrollo, almacenamiento, administración, localización y recuperación de todo tipo de contenido

digital (ADL, 2002). Existen varias definiciones sobre los ROA, entre ellas podemos citar las más importantes:

Según Sampedro, Sariago, Martínez, Martínez y Rodríguez (2005), los repositorios de objetos de aprendizaje se definen como “almacenes digitales en los que se recogen aportaciones individuales de los miembros de una comunidad para ser compartidos y evaluados entre todos ellos.”

Mientras que APROA (2006), dice que “Un repositorio de objetos de aprendizaje, corresponde a un "almacén o depósito" de Objetos de Aprendizaje (OA). Adicionalmente, provee de herramientas para la búsqueda de sus objetos de aprendizaje.”

Por otro lado, Varas en el 2002, sostiene que no es posible pensar en objetos de aprendizaje si no se los concibe albergados en repositorios. Como objetos aislados no tienen ninguna relevancia ni significado real. El tipo de componentes albergados en un repositorio, que deben tener sus propias identidades y ser por lo tanto localizables, son tan variados como gráficos, imágenes, textos, videos, documentos e integración de ellos como capítulos de un curso o hasta cursos completos. Un aspecto muy importante de los repositorios es que no necesariamente albergan físicamente los objetos que contienen; les basta con “apuntar” a ellos.

Considerando todo lo expuesto hasta este momento, se puede decir que los ROA son un tipo de bibliotecas digitales especializadas en recursos educativos que utilizan los estándares de metadatos que han desarrollado los organismos encargados de la estandarización del **e-learning**, preparadas tecnológicamente para interoperar con otros repositorios y con otras aplicaciones de los entornos **e-learning**. La interacción de los ROA (Ver Figura 1) debe ser posible con todos los sistemas, herramientas y usuarios que hagan uso de contenidos, así como con aquellos otros repositorios o recursos que pueda agregar a su catálogo o con los que pueda comunicarse para hacer posibles cualquier tipo de búsquedas.

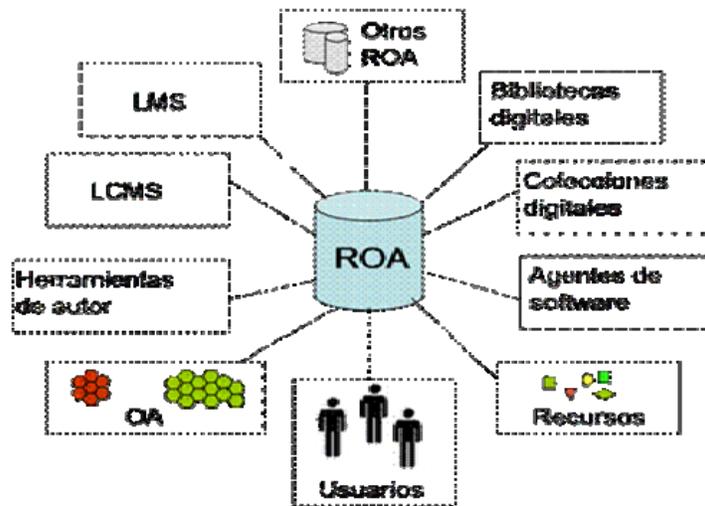


Figura 1. Interacción de los ROA

2.2.1. Tipos de ROA

2.2.1.1. Según la concentración de los recursos

Por la forma en la que se concentran los recursos, se identifican dos tipos de ROA (Downes, 2004) (Rehak & Mason, 2003):

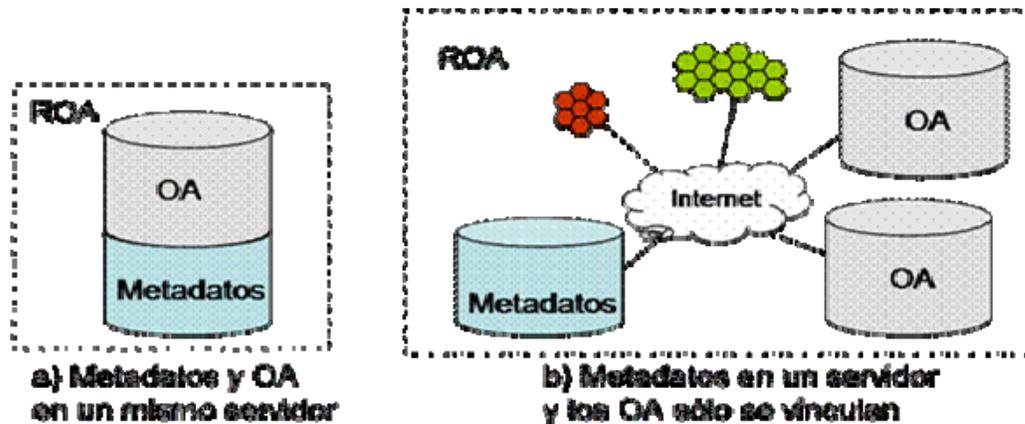


Figura 2. Tipos de ROA por la distribución de los OA

- Los que contienen los OA y sus metadatos, en éstos los objetos y sus descriptores se encuentran dentro de un mismo sistema e incluso dentro de un mismo servidor (ROA Locales) (Ver Figura 2a).

- 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 (ROA Distribuidos) (Ver Figura 2b).

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

2.2.1.2. Según la distribución de los metadatos

Los ROA más conocidos comúnmente funcionan de forma independiente (**stand-alone**). 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 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 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.

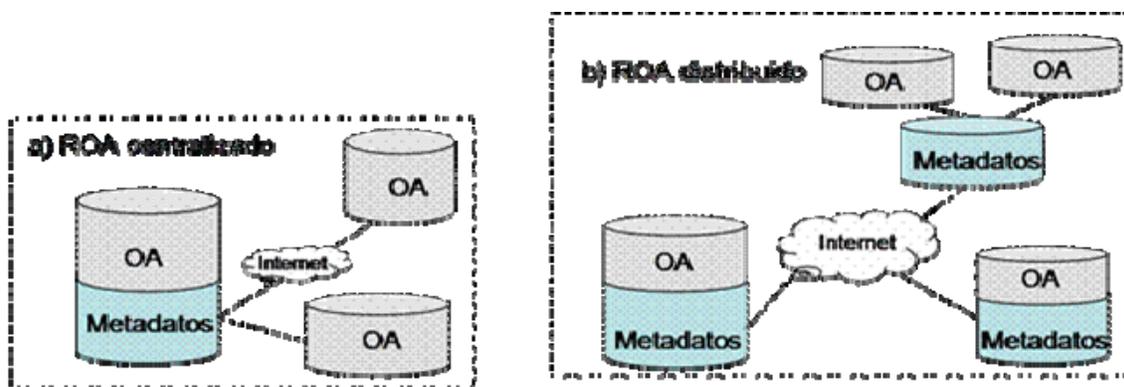


Figura 3. Tipos de ROA por la distribución de los metadatos.

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 (Ver Figura 3a), en los cuales metadatos de los OA están contenidos en un mismo servidor, aunque el objeto esté localizado en alguno otro. El modelo distribuido (Ver Figura 3b) opera a través de varios servidores, cada uno contiene diferentes grupos de metadatos y se comunican entre ellos para intercambiarlos.

2.2.2. Características de los Repositorios de Objetos de Aprendizaje

Los ROA al ser bibliotecas digitales deben cubrir al menos los mismos elementos básicos: colección, servicios de valor añadido, personalización y ciclo de vida. EduTools (Leslie, Landond, Lamb, & Poulin, 2004) realizó un estudio a productos de software para ROA y utilizó criterios divididos en 10 categorías, que expertos en la materia propusieron después de hacer una revisión exhaustiva de la literatura y de revisar y analizar los productos existentes en el mercado:

1. Herramientas de búsqueda. Considera la búsqueda a través de palabras clave u otros metadatos, la posibilidad de que el usuario pueda realizar exploraciones en listados predefinidos en alguna categorización o clasificación.
2. Herramientas de recopilación. Creación de **bookmarks** de recursos o colecciones personales y posibilidad de creación de paquetes con varios recursos.
3. Colectividad y evaluación. Posibilidad de que los usuarios puedan evaluar formal o informalmente un OA, mecanismos para registrar los diferentes contextos en los que el OA ha sido utilizado, y listas de OA que el usuario desearía se incluyeran o se modificaran.
4. Meta-etiquetado. Herramienta de etiquetado, soporte de estándares y/o varios esquemas, importación y exportación de metadatos.
5. Administración de contenidos. Seguimiento del flujo de creación y publicación de un OA, control de versiones y funciones de almacenamiento, herramientas de autoría.
6. Administración y cumplimiento de derechos digitales de autor. Registro, transmisión, interpretación y hacer cumplir los derechos de autor.
7. Presentación y salidas de consorcio. Accesibilidad, salidas en múltiples formatos para diferentes dispositivos, cambios de apariencia de la interfaz, soporte de caracteres de diferentes idiomas, transformación de formatos.
8. Integración e interoperabilidad. Federación y búsqueda en otros repositorios, integración con un administrador de cursos, soporte de servicios Web y de aplicaciones API (Application Program Interface) que puedan extraer información de actividades dentro del repositorio.
9. Consideraciones técnicas. Autenticación, autorización y personalización, informe de uso, soporte para diferentes sistemas operativos, especificaciones de la BD

requerida, escalabilidad, arquitectura del modelo de software, requisitos técnicos y humanos para su puesta en marcha.

10. Costo/licenciamiento/Otros. Información de la compañía que provee el software, número de instalaciones, modelo de costo o licenciamiento.

2.2.3. Funciones de los Repositorios

ADL (2002) propone un conjunto básico de **funciones que los repositorios** deben proveer a fin de dar acceso a los objetos de aprendizaje en un ambiente seguro. Estas funciones son:

- **Buscar/encontrar.** Es la habilidad para localizar un OA apropiado. Esto incluye la habilidad para su despliegue.
- **Pedir.** Un OA que ha sido localizado.
- **Recuperar.** Recibir un OA que ha sido pedido.
- **Enviar.** Entregar a un repositorio un OA 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.

2.2.4. Ejemplos de Repositorios de Objetos de Aprendizaje

A continuación se explican algunos de los distintos Portales de Repositorios más utilizados:

- **MERLOT** (Multimedia Educational Resource for Learning and Online Teaching):
Es el repositorio más conocido y reconocido, es quizá el que está marcando la pauta para el desarrollo y tendencia de los ROA. Es un repositorio centralizado que contiene sólo los metadatos y apunta a los objetos ubicados en sitios remotos. Es independiente y funciona como un portal de OA. Provee búsquedas y otros servicios como personalización, importación y exportación de objetos. Disponible en <http://www.merlot.org/>

- **CAREO** (Campus Alberta Repository of Educational Objects):

Es un repositorio centralizado de OA multidisciplinares de profesores de Alberta (Canadá). Es un repositorio independiente que da acceso a objetos remotos y locales a través de los metadatos contenidos en su colección. Cualquier usuario puede tener acceso a los objetos, pero los miembros tienen servicios adicionales, al igual que MERLOT ser miembro es gratis y abierto a cualquier persona. Disponible en <http://www.careo.org/>

- **CLOE** (Co-operative Learning Object Exchange):

Es un modelo cooperativo para el desarrollo, uso y reutilización de OA. Es un proyecto de la Universidad de Waterloo en el que participan 17 universidades de Ontario y se tiene acceso a su colección sólo siendo miembro de dichas universidades. Disponible en <http://cloe.on.ca/>

2.3. Estándares para Objetos de Aprendizaje

2.3.1. SHARABLE CONTENT OBJECT REFERENCE MODEL (SCORM)

Según su visión, ADL desde su posición de liderazgo debida al respaldo de la Administración Norteamericana, propuso el modelo SCORM con el objetivo de establecer un marco común para el aprendizaje asistido por computadora y basado en la red Internet. Este marco común provee un conjunto de guías, especificaciones y estándares basados en las especificaciones previamente existentes en el campo propuestas por distintas organizaciones.

SCORM ha tenido y tiene un gran impacto en el campo del aprendizaje a través de Internet dado que tanto la industria como el mundo académico han reconocido el liderazgo de ADL como entidad de referencia a la hora de valorar la calidad de los procesos de aprendizaje. En la actualidad la compatibilidad con SCORM es el principal punto de encuentro entre todas las organizaciones implicadas en el campo del aprendizaje asistido por computadora. El peso de ADL ha convertido a SCORM en un requisito prácticamente indispensable de cara a la comercialización de un nuevo producto de enseñanza.

La definición del modelo SCORM, así como su evolución y las distintas decisiones de diseño tomadas durante el proceso de especificación, se basan en 6 principios esenciales que en la visión de la Iniciativa ADL se enuncian como:

- **Accesibilidad.** Definida como la posibilidad de localizar y acceder a componentes instruccionales desde una ubicación remota y su envío a otras muchas localizaciones.
- **Adaptabilidad.** Definida como la posibilidad de adaptar la enseñanza a distintas necesidades individuales u organizacionales.
- **Asequibilidad.** Definida como la posibilidad de aumentar la eficiencia y la productividad reduciendo el tiempo y el coste invertidos en la enseñanza.
- **Durabilidad.** Definida como la posibilidad de resistir la evolución de la tecnología y futuros cambios sin incurrir en rediseños, reconfiguraciones o recodificaciones excesivamente costosas.
- **Interoperabilidad.** Definida como la posibilidad de tomar componentes instruccionales desarrollados en una ubicación determinada y empleando unas herramientas y plataformas determinadas para su posterior aplicación en otra ubicación y otro conjunto de herramientas y plataformas.
- **Reusabilidad.** Definida como la flexibilidad para incorporar componentes instruccionales en múltiples contextos y aplicaciones.

2.3.2. LEARNING OBJECT METADATA (LOM)

Los metadatos son información añadida a los materiales digitales que facilitan su clasificación y posterior recuperación. La especificación de metadatos adecuados para los materiales educativos es indispensable a fin de añadir valor a los mismos, en el sentido de facilitar su reutilización. Efectivamente, los materiales enriquecidos convenientemente con metadatos podrán almacenarse en bibliotecas digitales de contenidos educativos (por ejemplo, *repositorios* de objetos de aprendizaje). Estas bibliotecas soportarán, entonces, consultas significativas que permitirán la recuperación de aquellos materiales almacenados que cubran una determinada necesidad pedagógica.

La utilidad de un esquema de metadatos radica en su aceptación por una comunidad suficientemente amplia de productores y consumidores de material educativo. Efectivamente, si dos comunidades utilizan esquemas de metadatos distintos, difícilmente

los materiales producidos podrán coexistir en un mismo repositorio, a menos que se haya encontrado previamente un consenso que permita *homogeneizar* los metadatos utilizados por ambas comunidades (por ejemplo, transformándolos a un esquema común).

Es por ello que, desde la comunidad de e-learning, se han realizado distintos esfuerzos para estandarizar los esquemas de metadatos que deben ser utilizados en la producción de contenidos educativos. El esfuerzo más prometedor ha desembocado en el estándar IEEE LOM (del inglés, *Learning Object Metadata*) (ver IEEE LOM 2002).

Según ADL en el 2007, LOM es un estándar que tiene como objetivo guiar en el mercado de recursos educativos para potenciar su búsqueda, evaluación, obtención y utilización, es decir, establece un modelo de datos utilizado por SCORM, usualmente codificado en XML, que se utiliza para describir un OA. De esta forma, se facilita la reutilización de OA y su interacción.

Una Visión Conceptual de la Adopción de LOM

Los metadatos en la adopción del estándar LOM están agrupados en *categorías de metadatos*. Más concretamente, LOM distingue 9 categorías de metadatos diferentes:

- *Categoría general*. Los metadatos en esta categoría representan información general sobre el material educativo que describe el mismo como un todo.

Categoría	Elementos de metadatos
1.- General	1.1 Identificador
	1.2 Título
	1.3 Idioma del Objeto
	1.4 Descripción
	1.5 Palabras Claves
	1.6 Ámbito
	1.7 Estructura
	1.8 Nivel de Agregación

- *Categoría lifecycle (ciclo de vida)*. Esta categoría agrupa metadatos referidos a la historia y estado actual del proceso de producción y mantenimiento del material educativo por parte de los autores.

Categoría	Elementos de metadatos
2.- Ciclo de Vida	2.1 Versión
	2.2 Estado
	2.3 Contribución

- *Categoría metametadata* (meta-metadatos). Esta categoría agrupa información relativa a los metadatos en sí (de ahí su nombre).

Categoría	Elementos de metadatos
3.- Meta-metadatos	3.1 Identificador
	3.2 Contribución
	3.3 Esquema de Metadatos
	3.4 Idioma

- *Categoría technical* (técnica). Categoría que agrupa metadatos relativos a las características y requisitos técnicos del material en sí.

Categoría	Elementos de metadatos
4.- Requisitos Técnicos	4.1 Formato
	4.2 Tamaño
	4.3 Localización
	4.4 Requisitos
	4.5 Pautas de Instalación
	4.6 Otros requisitos de Plataforma
	4.7 Duración

- *Categoría educational* (educativa). Categoría que agrupa metadatos relativos a los usos educativos del material.

Categoría	Elementos de metadatos
5.- Uso Pedagógico	5.1 Tipo de interactividad
	5.2 Tipo de recurso educativo
	5.3 Nivel de interactividad
	5.4 Densidad semántica
	5.5 Destinatario
	5.6 Contexto
	5.7 Rango típico de edad
	5.8 Dificultad
	5.9 Tiempo típico de aprendizaje
	5.10 Descripción
	5.11 Idioma

- *Categoría rights* (derechos). Categoría que agrupa metadatos relativos a los derechos de propiedad e intelectuales del material.

Categoría	Elementos de metadatos
6.- Derechos de uso	6.1 Coste
	6.2 Derecho de autor y otras restricciones
	6.3 Descripción

- *Categoría relation* (relación). Categoría de metadatos utilizados para establecer relaciones entre el material y otros materiales.

Categoría	Elementos de metadatos
7.- Relaciones	7.1 Tipo
	7.2 Recurso
	7.3 Descripción

- *Categoría annotation* (anotación). Anotaciones y comentarios sobre el material educativo.

Categoría	Elementos de metadatos
-----------	------------------------

8.- Anotación	8.1 Entidad
	8.2 Fecha
	8.3 Descripción

- *Categoría classification* (clasificación). Metadatos para la clasificación del material en taxonomías.

Categoría	Elementos de metadatos
9.- Clasificación	9.1 Propósito
	9.2 Ruta Taxonómica
	9.3 Descripción
	9.4 Palabras claves

2.4. Proyecto AMBAR

AMBAR es un proyecto donde se busca crear un sistema generador de ambientes constructivistas de enseñanza-aprendizaje, es decir, un sistema capaz de servir de herramienta de ayuda a profesores, estudiantes y demás usuarios, para la enseñanza y/o aprendizaje de cualquier tema académico. Usando como vía de comunicación la Web, para facilitar el esparcimiento de la información y servir de esta manera a la expansión de conocimiento.

SIG-AMBAR, Siglas de Sistema generador de **AMB**bientes constructivistas de enseñanza-**ApR**endizaje, es un proyecto de investigación y desarrollo de la Escuela de Computación, Facultad de Ciencias, Universidad Central de Venezuela. Este proyecto busca proporcionar una plataforma tecnológica que soporte el almacenamiento, generación, uso y reuso de OA y Diseños de Aprendizaje en ambientes instruccionales bajo enfoques cognitivos constructivistas, que a su vez sean compatibles con los estándares actuales IMS LD y SCORM, y que provean los servicios de Web Semántica. (Miguel, López y Montaña, 2006).

2.4.1. Visión Interdisciplinaria de AMBAR

La idea de una visión interdisciplinaria de AMBAR es incluir distintas perspectivas para su desarrollo (Miguel, López y Montaña, 2006):

La primera perspectiva es desde la ingeniería del software, que involucra una visión interna que aporta conocimiento en todo lo referente al desarrollo del software; y

una visión externa que involucra todo el conocimiento en lo que es la comunicación con los usuarios (aprendices y docentes) o lo que es bien llamado Interacción Humano Computador (IHC).

La segunda perspectiva es la referente a la plataforma tecnológica que soportará a AMBAR, en donde estará representada la base de datos que permitirá almacenar, utilizar y compartir el conocimiento entre los usuarios.

La tercera perspectiva trata sobre el diseño instruccional que involucra desde el desarrollo del software hasta la generación de ambientes de enseñanza – aprendizaje por el sistema.

En la Figura (Ver Figura 4) se muestra una visión de las disciplinas consideradas en el proceso de desarrollo de AMBAR como sistema generador y al mismo tiempo como ambiente constructivista de enseñanza aprendizaje.

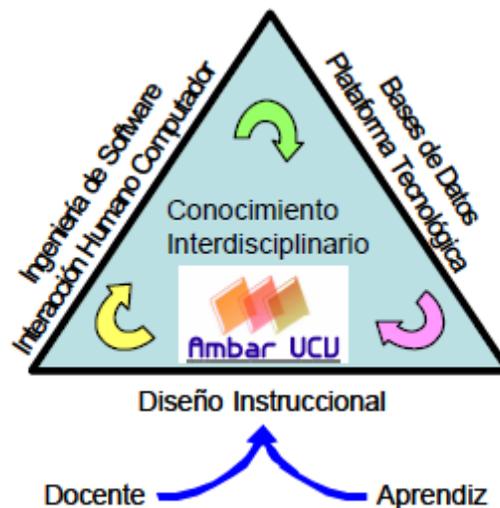


Figura 4. Estructura Interdisciplinaria de AMBAR.

“Una visión Interdisciplinaria del Sistema Generador de AMBientes de Enseñanza -ApRendizaje Constructivistas basados en Objetos de Aprendizaje (AMBAR)” López, M., Montaña, N., Miguel V. Congreso de Humanidades 2006.

2.4.2. Características de AMBAR

AMBAR permitirá a profesores y aprendices elaborar y participar en procesos de enseñanza aprendizaje constructivistas basados en OA reusables. Está constituido por tres ambientes: un ambiente para facilitarle al docente el diseño instruccional, un

ambiente para la colaboración entre estudiantes y docentes para la generación de conocimiento (basado en un modelo pedagógico) y un módulo para el mantenimiento del repositorio de OA y todos los demás elementos asociados al ambiente.

A nivel genérico, los tipos de OA que se manejan en AMBAR son:

- OA nivel micro (OA Fundamentales): representan contenido independiente del contexto y pueden ser usados para poblar estrategias y artefactos generados por los aprendices. Este tipo de objetos puede incluir gráficos, videos, definiciones, explicaciones descontextualizadas o lecturas, casos particulares y definición de problemas, entre otros.
- Objetos de Información Combinada (OA Generativo-Instruccionales): pudieran existir en un continuo desde un nivel micro hasta un nivel macro desde contenido con un contexto mínimo añadido (por ejemplo, enlaces dentro de un caso de estudio a perspectivas y temas, pudiendo incluir metas de aprendizaje) hasta actividades de aprendizaje más completas o estrategias instruccionales como tutoriales, micromundos, simulaciones, etc. Estos objetos también pudieran poblar los estrategias y artefactos de los aprendices o podrían ofrecerse como ayuda o guía.
- Estrategias (andamiaje a nivel macro): estos objetos están contextualizados por la implementación de enfoques instruccionales específicos (hipertextos cognitivos flexibles, aprendizaje basado en problemas, etc.) y pueden incorporar otros OA y varias clases de enlaces. La estrategia provee el contexto o la estructura para el aprendiz y está definido como un objeto dentro de la base de datos.

AMBAR necesita proveer y considerar ciertas características para incorporar principios constructivistas (capacidad de almacenar, usar, compartir diseños instruccionales y que además puedan ser soportados por diferentes herramientas y plataformas), soportar actividades de aprendizaje generativo (capacidad de que los aprendices puedan crear artefactos con diferentes niveles de granularidad y convertirlos en OA), hacer más flexible el contenido de los objetos y soportar las actividades de un ambiente de aprendizaje constructivista (capacidad de considerar diferentes OA como los fundamentales, los combinados y los frameworks).

A partir de las características deseadas para AMBAR, se puede observar en la Figura 5 el diagrama de clases que describe la estructura a nivel macro del repositorio de OA, donde se manejan los tres tipos de OA genéricos mencionados anteriormente (fundamentales, combinados y frameworks). (López, Montaña y Miguel, 2005)

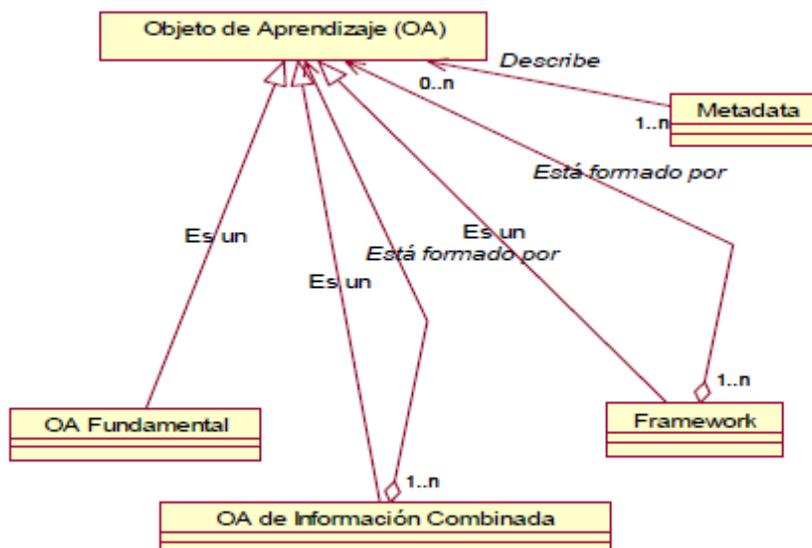


Figura 5. Diagrama de Clases a nivel macro del repositorio de OA

(Fundamentales, de Información Combinada y Frameworks). "Sistema Generador de AMBientes de Enseñanza-ApRendizaje Constructivistas basados en Objetos de Aprendizaje (AMBAR)" López, M., Montaña, N., Miguel V. Universidad Central de Venezuela (UCV) (2005).

En la Figura 6 se muestra el Diagrama de Casos de Uso Principal de AMBAR, en la cual se reflejan los tipos de usuarios del sistema, las funcionalidades principales que debe proveer AMBAR y la relación de cada tipo de usuario con el mismo. La descripción de la funcionalidad de cada caso de uso es la siguiente (López, Montaña y Miguel, 2005):

- **Registrarse en AMBAR:** Ésta funcionalidad permite crear usuarios registrados en el sistema del tipo profesor o aprendiz.
- **Generar OA:** Ésta funcionalidad le permite a los profesores o aprendices generar OA nuevos o por agregaciones y combinaciones de objetos existentes en el repositorio.
- **Generar nuevas versiones de OA:** Ésta funcionalidad le permite a los dueños de los OA generar nuevas versiones.

- **Almacenar OA en el repositorio:** Ésta funcionalidad permite a los usuarios autorizados almacenar OA al repositorio.
- **Etiquetar OA:** Ésta funcionalidad permite añadir metadatos a los OA existentes en el repositorio, cumpliendo con los estándares existentes.
- **Generar Ambientes de Aprendizaje:** Ésta funcionalidad permite generar ambientes de aprendizaje constructivistas en base a OA existentes en el repositorio.
- **Limpiar:** Ésta funcionalidad le permite a los usuarios registrados poder borrar del repositorio OA que ya son obsoletos o que han sido descartados por sus dueños.
- **Exportar OA:** Ésta funcionalidad permite que los OA existentes puedan ser exportados a otros sistemas manejadores de aprendizaje o a otras herramientas de diseño y edición.
- **Importar OA:** Ésta funcionalidad permite importar OA existentes en otros sistemas manejadores de aprendizaje o en otras herramientas de diseño y edición.

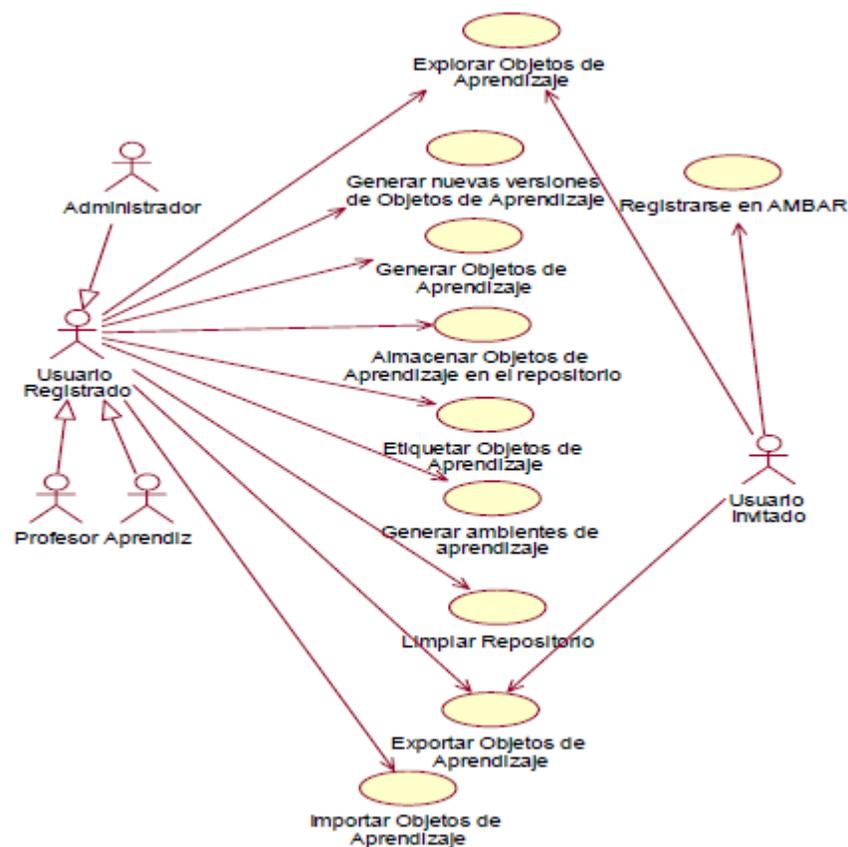


Figura 6. Diagrama de Casos de Uso (funcionalidades principales y sus usuarios potenciales) de AMBAR.

El repositorio de AMBAR es el lugar donde están albergados todos los OA creados y/o utilizados por los diferentes ambientes de enseñanza-aprendizaje del generador de AMBientes de enseñanza- ApRendizaje constructivista (AMBAR). (López, Miguel y Montaña, 2006). Las principales funcionalidades del repositorio de AMBAR tienen que ver con las características propias de AMBAR, es decir, el permitir almacenar, consultar, usar y reutilizar los OA producidos o usados en los ambientes de aprendizaje, y que éstos OA puedan ser soportados por diferentes plataformas y por diferentes herramientas, que el etiquetamiento sean de acuerdo a los estándares actuales, el manejar múltiples versiones de OA, etc; son parte de las características de AMBAR.

CAPÍTULO 3. ANÁLISIS SOBRE LA BASE DE DATOS Y EL REPOSITORIO DE METADATA DE AMBAR

Para el desarrollo de este trabajo se realizó un análisis sobre ambos proyectos que se iban a integrar, de manera que se tuviera claro el ambiente de desarrollo en el cual ambos están implementados y el código utilizado. Luego del resultado del análisis, y preparado el ambiente de desarrollo, teniendo en cuenta todos los aspectos relacionados con cada uno de los proyectos, se procedió a realizar el proceso de integración a través del método de desarrollo propuesto.

A continuación se describen los proyectos que se van a integrar:

1. Base de Datos de AMBAR

La base de datos de AMBAR, se desarrolló a través de la Capa de Servicios Web, como implementación de una Arquitectura Orientada a Servicios (SOA). La definición de los Servicios Web para el acceso a la base de datos de AMBAR, mejora la interoperabilidad entre las diferentes aplicaciones que constituyen el sistema, ya que éstos ofrecen una alternativa de software independiente en cuanto a la plataforma, basada en estándares para la integración de aplicaciones, la automatización de procesos de negocio y la publicación de la información de diversas fuentes.

El sistema proporciona un espacio para manipular OA que pueden ser almacenados y recuperados de la base de datos de AMBAR, a través del Sistema Manejador de Base de Datos Orientado a Objetos Database for objects (db4o) de software libre, basado en una arquitectura MVC (Modelo Vista Controlador) para los Servicios Web y haciendo uso del patrón DAO (Data Access Object) para el acceso a los datos. Los objetos son descritos a través de una metadata asociada, que permite identificarlos para luego potenciar su búsqueda, evaluación, obtención y reutilización, es decir, establece un modelo de datos utilizado por SCORM, que no es más que un conjunto de normas técnicas que permiten a los sistemas de administración de aprendizaje (LMS, por sus siglas en inglés) en línea, importar y reutilizar contenidos de aprendizaje que se ajusten al estándar.

estructuradas, las cuales a su vez pueden formadas por otras actividades estructuradas, y por actividades simples que a su vez están compuestas por OA (Fundamentales y/o Información combinada) y por servicios (chats, foros, etc.) que permiten realizar actividades individuales, colaborativas, presenciales o a distancia.

Los servicios que están definidos para este modelo conceptual son los siguientes:

Objetos de aprendizaje (OA)

- Almacenar OA: consiste en almacenar un OA Fundamental o de Información Combinada en la base de datos. Describiéndolos a través de la metadata asociada.
- Buscar OA: consiste en localizar un OA almacenado en la base de datos de dos maneras: por palabra clave definida en la metadata asociada o búsqueda de todos los OA almacenados en la base de datos.
- Modificar OA: consiste en modificar la descripción del OA a través de la clase metadata asociada al mismo.
- Eliminar OA: consiste en eliminar un OA de la BD de AMBAR, eliminando también todas las asociaciones que pueda tener con otra clase existente.

Actividad

- Almacenar Actividad: consiste en almacenar una actividad simple o estructurada en la BD. Describiéndolas a través de la metadata asociada, que se integra de la primera iteración.
- Buscar Actividad: consiste en localizar una actividad almacenada en la BD de dos maneras: por palabra clave definida en la metadata asociada o búsqueda de todas las actividades almacenadas en la BD.
- Modificar Actividad: consiste en modificar la descripción de la actividad a través de la clase metadata asociada.
- Eliminar Actividad: consiste en eliminar una actividad de la BD de AMBAR, eliminando también todas las asociaciones que pueda tener con otra clase existente.
- Agregar Actividad: consiste en realizar todas las asociaciones que tiene la clase actividad (actividad simple o actividad estructurada) con otras clases.

Servicio

- Almacenar Servicio: consiste en almacenar un servicio en la BD.
- Buscar Servicio: consiste en localizar un servicio almacenado en la BD de dos maneras: por el nombre o búsqueda de todos los servicios almacenados en la BD.
- Modificar Servicio: consiste en modificar los atributos que describen el servicio.
- Eliminar Servicio: consiste en eliminar un servicio de la BD de AMBAR, eliminando también todas las asociaciones que pueda tener con otra clase existente.

Ambiente

- Almacenar Ambiente: consiste en almacenar un ambiente de aprendizaje en la BD. Describiéndolo a través de la metadata asociada, que se integra de la segunda iteración.
- Buscar Ambiente: consiste en localizar un ambiente de aprendizaje almacenado en la BD de dos maneras: por palabra clave definida en la metadata asociada o búsqueda de todos los ambientes de aprendizaje almacenados en la BD.
- Modificar Ambiente: consiste en modificar la descripción del ambiente de aprendizaje a través de la clase metadata asociada.
- Eliminar Ambiente: consiste en eliminar un ambiente de aprendizaje de la BD de AMBAR, eliminando también todas las asociaciones que pueda tener con otra clase existente.
- Agregar Ambiente: consiste en realizar todas las asociaciones que tiene la clase ambiente de aprendizaje con otras clases.

Participante

- Almacenar Participante: consiste en almacenar un participante en la BD.
- Buscar Participante: consiste en localizar un participante almacenado en la BD de dos maneras: por atributos nombre, apellido, usuario, organización o búsqueda de todos los participantes almacenados en la BD.
- Modificar Participante: consiste en modificar los atributos de un participante.
- Eliminar Participante: consiste en eliminar un participante de la BD de AMBAR, eliminando también todas las asociaciones que pueda tener con otra clase existente.

Rol

- Almacenar Rol: consiste en almacenar un rol en la BD.

- Buscar Rol: consiste en localizar un rol almacenado en la BD búsqueda de todos los roles almacenados en la BD.
- Modificar Rol: consiste en modificar los atributos de un rol.
- Eliminar Rol: consiste en eliminar un rol de la BD de AMBAR, eliminando también todas las asociaciones que pueda tener con otra clase existente.
- Agregar Rol: consiste en realizar todas las asociaciones que tiene la clase rol con otras clases.

Método

- Almacenar Método: consiste en almacenar un método en la BD.
- Buscar Método: consiste en localizar un método almacenado en la BD de dos maneras: por atributo nombre o búsqueda de todos los métodos almacenados en la BD.
- Modificar Método: consiste en modificar los atributos de un método.
- Eliminar Método: consiste en eliminar un método de la BD de AMBAR, eliminando también todas las asociaciones que pueda tener con otra clase existente.

Competencia

- Almacenar Competencia: consiste en almacenar una competencia en la BD.
- Buscar Competencia: consiste en localizar una competencia almacenada en la BD de dos maneras: por atributo nombre o búsqueda de todos las competencia almacenadas en la BD.
- Modificar Competencia: consiste en modificar los atributos de una competencia.
- Eliminar Competencia: consiste en eliminar una competencia de la BD de AMBAR, eliminando también todas las asociaciones que pueda tener con otra clase existente.
- Agregar Competencia: consiste en realizar todas las asociaciones que tiene la clase competencia con otras clases.

Prerequisitos

- Almacenar Prerequisitos: consiste en almacenar los prerequisitos en la BD.
- Buscar Prerequisitos: consiste en localizar los prerequisitos almacenados en la BD por búsqueda de todos los prerequisitos almacenados en la BD.

- Modificar Prerequisitos: consiste en modificar los atributos de los prerequisitos.
- Eliminar Prerequisitos: consiste en eliminar los prerequisitos de la BD de AMBAR, eliminando también todas las asociaciones que pueda tener con otra clase existente.
- Agregar Prerequisitos: consiste en realizar todas las asociaciones que tiene la clase prerequisitos con otras clases.

2. Repositorio de Metadata

El repositorio de metadata, permite la organización estructurada de los atributos que describen a los OA, facilitando su búsqueda, recuperación y administración. Este repositorio es de vital importancia para AMBAR pues actualmente no existen herramientas que provean de estas facilidades; teniendo que recurrir a mecanismos no funcionales para tales efectos.

Es importante destacar que la metadata de los OA obedece a las especificaciones de un estándar mundial especializado para describirla llamado Learning Object Metadata (LOM), el cual está contenido dentro de una especificación llamada Sharable Content Object Reference Model (SCORM), la cual fue diseñada específicamente para estructurar este tipo de objetos. Esto hace que el repositorio sea compatible e interoperable con los sistemas de gestores de aprendizaje.

Por otro lado el repositorio donde se almacena la Metadata, fue diseñado sobre un sistema manejador de base de datos orientado a objetos (SMBDOO) llamado db4o, el cual es accedido mediante una capa Web basada en Servicios. Luego esta capa de servicios es la que finalmente es usada por la capa de aplicación, con la que interactúan los usuarios finales. Visto de esta manera, el sistema ofrece la posibilidad de abstracciones e independencia entre los diferentes factores que conforman el repositorio de metadata, dando a su vez interoperabilidad con otros sistemas y la tecnología necesaria para facilitar los tiempos de acceso.

La interfaz de usuario final se basó en la tecnología de páginas web dinámicas cliente-servidor (JSP: Java Server Pages), que interactúa con todos los métodos de acceso a los datos que hacen abstracción e independencia de la BD propiamente; esto

les permite a los usuarios finales, manipular la Metadata que describe a los OA almacenados en el repositorio de objetos de AMBAR.

Tanto el repositorio, la aplicación web, como los métodos de acceso fueron desarrollados con una de las premisas de AMBAR: software libre. Entre ellas podemos mencionar: db4o, como manejador de base de datos orientado a objetos. NetBeans, como ambiente de desarrollo Java para elaborar los métodos de acceso al repositorio. Java Servlet, como ambiente de servidor para construir las páginas WEB. JSP (Java Server Pages), como lenguaje de desarrollo de las páginas web que conforman la interfaz de usuario. Apache Tomcat, como servidor de aplicaciones y despliegue Web. Axis, como implementación de Simple Object Access Protocol (SOAP) para la intercomunicación de los servicios web y sus clientes. Mozilla Firefox, como navegador web para realizar las pruebas. JavaMail y JXMSEC, entre otras librerías de java para facilitar el desarrollo de las implementaciones.

Los resultados de este trabajo consisten en la BD de metadata basada en el estándar LOM, los servicios web necesarios para accederla y una aplicación que a través del uso de los servicios web definidos permite interactuar con el repositorio de metadata de AMBAR, respondiendo a las necesidades de los usuarios del sistema en cuanto a la búsqueda, recuperación y gestión de metadata de OA.

En la Figura 8 se muestra el Diagrama Conceptual de la BD de Metadata:

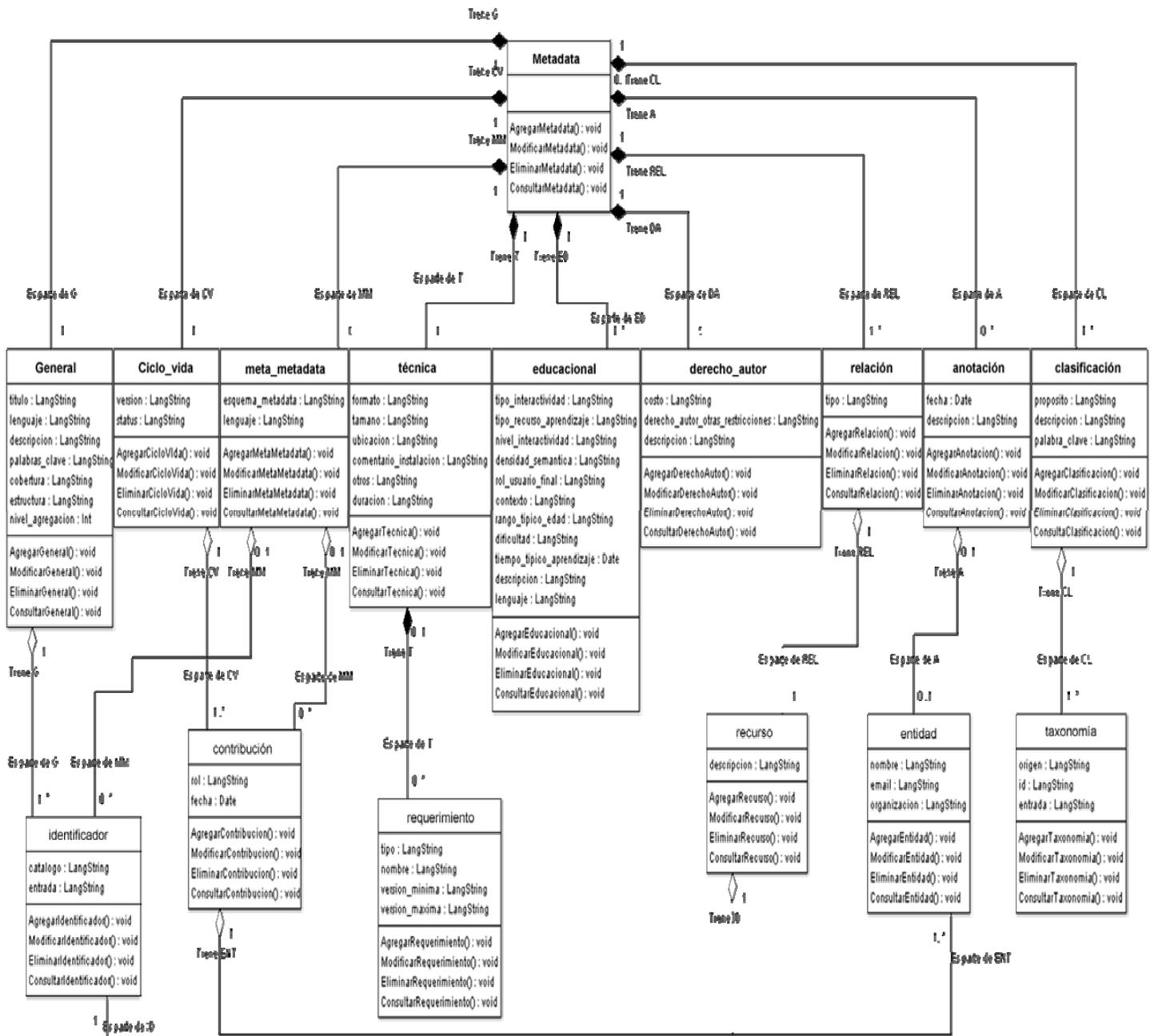


Figura 8. Diagrama conceptual de la Metadata.

Se puede observar que existe una clase raíz llamada Metadata, el cual posee todos los métodos asociados a la metadata (agregar, consultar, modificar y eliminar). Seguidamente, esta clase se relaciona directamente con otras nueve (9) clases; las cuales representan las nueve (9) categorías de LOM (General, Ciclo de Vida, Meta-metadata, Técnica, Educativa, Derecho de Autor, Relación, Anotación y Clasificación). Las categorías pueden o no tener relación con otros objetos aparte de la Clase Metadata,

ya que algunos de los atributos de las categorías pueden ser representados por Clases, las cuales son: Identificador, Contribución, Entidad, Requerimiento, Recurso y Taxonomía.

LOM estandariza con estas nueve categorías la información que describe a un OA, pero para que dicha información esté representada en una estructura mantenga su integridad, se crearon relaciones fuertes y débiles según fuera el caso entre cada uno de sus integrantes. De esta forma, es posible recorrer la estructura sin perder la conexión que existe entre cada uno de sus componentes. Las relaciones fuertes se dan sobre sólo dos elementos y las relaciones débiles se dan sobre dos o más elementos.

Los métodos para interactuar con el repositorio de metadata fueron los siguientes:

1. Agregar Metadata: El cual se encarga de recibir desde la aplicación Web toda la metadata de un OA y convertirla en un objeto con estructura LOM para luego instanciarlo y almacenarlo en el repositorio.
2. Buscar Metadata: Cuya función principal es traer toda la metadata asociada aun OA basado en tres tipos de criterios: Por título del objeto, por contexto y por propósito.
3. Modificar Metadata: Consiste en modificar elemento(s) de la metadata de un OA en específico.
4. Eliminar Metadata: Consiste en eliminar una metadata en el repositorio. Lo cual consiste en eliminar la persistencia de un objeto en el repositorio de metadata.

Servicios desarrollados:

1. agregarMD
 - Descripción: Servicio que se encarga de agregar metadata en el repositorio
 - Entrada: 63 parámetros con toda la información de un OA.
 - Salida: Un String que contiene el resultado de la operación (satisfactorio o fallido).
2. busquedaBasica
 - Descripción: Se encarga de realizar una búsqueda en el repositorio de metadata.

- Entrada: parámetro con la palabra a buscar y un segundo parámetro con el nombre del campo por el cual se hará el descenso y búsqueda en la BD.
 - Salida: Una matriz de dos dimensiones de Strings, donde la primera dimensión representa los objetos encontrados y la segunda dimensión representa los elementos de cada OA recuperado.
3. eliminarPorOID
- Descripción: Se encarga de eliminar un OA en el repositorio de metadata.
 - Entrada: Variable tipo Long que contiene el identificador generado por el manejador db4o referente al objeto almacenado.
 - Salida: Un String que contiene el resultado de la operación (satisfactorio o fallido).
4. busquedaPorOID
- Descripción: Se encarga de realizar una búsqueda en el repositorio de metadata.
 - Entrada: Variable tipo Long que contiene el identificador generado por el manejador db4o referente al objeto almacenado.
 - Salida: Una matriz de String que contiene los elementos del OA recuperado.
5. busquedaAvanzada
- Descripción: Se encarga de realizar una búsqueda en el repositorio de metadata.
 - Entrada: arreglo de String de 20 posiciones que contiene todos los parámetros por los cuales se buscará el OA.
 - Salida: Una matriz de dos dimensiones de Strings, donde la primera dimensión representa los objetos encontrados y la segunda dimensión representa los elementos de cada OA recuperado.

Es importante destacar que debido a que el repositorio de Metadata sólo almacena metadatos de OA, la integración es enfocada específicamente en la sección de OA del modelo conceptual de la BD del SIG-AMBAR.

CAPÍTULO 4. MARCO METODOLÓGICO

Dando continuidad al desarrollo de este trabajo y para lograr los objetivos planteados, en este capítulo se llevará a cabo la aplicación del método de desarrollo propuesto en el capítulo uno.

Inicialmente en la investigación, debido a que se necesitaba integrar dos elementos existentes del proyecto AMBAR, se analizaron ambos repositorios para ver cuál era el puente de conexión para integrarlos, lo cual resultó sencillo ya que ambos elementos, fueron desarrollados de forma tal que fuera fácil su acoplamiento, lo que garantizó la compatibilidad entre los mismos.

Debido a que ambos proyectos ya estaban implementados y no se requirió todo el diseño inicial de implementación, se propuso que dentro las 4 fases del método de desarrollo, las fases de Elaboración y Construcción sean una misma, llamada fase de Integración, basándose finalmente este método de desarrollo en 3 fases, las cuales son: Inicio, Integración y Pruebas.

Sobre las fases planteadas, se proponen 2 iteraciones, donde el número de iteraciones se determinaron agrupando aquellas funcionalidades que eran comunes, para de esta forma tener una mejor planificación de los cambios que se iban a realizar. Dichas iteraciones contendrán a grandes rasgos lo siguiente:

- **Iteración 1:** Adecuación del repositorio de Metadata, tanto a nivel funcional como la interfaz de usuario, de manera que se pueda integrar con la base de datos de AMBAR a través de la Capa de Servicios.
- **Iteración 2:** Reestructuración de la Capa de presentación y de los enlaces correspondientes al almacenamiento, modificación y eliminación de OA de la Capa de Servicios Web de la base de datos de AMBAR para que utilicen los servicios correspondientes, a cada uno de ellos, del Repositorio de Metadata.

A continuación se describen las fases desarrolladas:

Fase de Inicio

Esta fase consiste en determinar el alcance del desarrollo y la comprensión el análisis de los requerimientos del sistema, para así obtener un buen funcionamiento de la integración que se desea realizar. Para llevar a cabo las tareas relacionadas con esta fase se realizó lo siguiente:

- Estudio de la iteración anterior a la actual, con el fin de realizar el acoplamiento y las adaptaciones necesarias para el buen funcionamiento de la siguiente iteración de desarrollo.
- Actualización y corrección de nuevos requerimientos de usuario.
- Se definieron las nuevas clases a ser implementadas.
- Se definieron los cambios necesarios sobre los diagramas ya modelados de ambos sistemas (Casos de uso, modelo conceptual y nueva arquitectura)
- Diseño de los cambios, a nivel de interfaz de usuario, de las aplicaciones Web a integrar, que permitirán realizar las pruebas unitarias e integrales.

Fase de Integración

En esta fase se elabora la codificación de un producto operativo, tomando en cuenta el diseño que se elaboró en la fase anterior. Para llevar a cabo las tareas relacionadas con esta fase se realizó lo siguiente:

- Se mantuvieron las plataformas utilizadas inicialmente por los dos sistemas.
- Se crearon los elementos que permitieron la integración de ambas aplicaciones
- Se modificaron algunos códigos de manera que se ajustaran a la integración y que fuera más fácil su entendimiento, tanto para su modificación como para futuros mantenimientos o nuevas versiones.
- Se modificaron todas las interfaces de usuario, para ajustarlas a la incorporación del modelo del Repositorio de Metadata.
- Se integró el código realizado en la iteración 1 a la iteración siguiente.

Fase de Transición

Esta fase consiste en implementar el producto en el usuario final, completando toda la documentación requerida por el usuario. Como consecuencia de esto suelen surgir nuevos requisitos para ser analizados, los cuales se deben contemplar e incluir en la siguiente iteración de trabajo.

Para llevar a cabo las tareas relacionadas con esta fase se realizó lo siguiente:

- Inducción a los usuarios del funcionamiento de la aplicación integrada.
- Verificación de consistencia de la integración.
- Realización de las correcciones necesarias de acuerdo a las especificaciones realizadas por el usuario.
- Realización de pruebas unitarias e integrales que permitieron realizar los ajustes y recomendaciones para futuras nuevas implementaciones.

El desarrollo de este trabajo estará basado en el modelo conceptual de base de datos de AMBAR (Ver Figura 7) y el modelo conceptual del Repositorio de Metadatos (Ver Figura 8), donde se observará la integración de ambos a lo largo de cada fase e iteración de este trabajo.

4.1. Primera Iteración

En esta primera iteración, se definieron las tareas a ser realizadas en el Repositorio de Metadatos, tales como:

- Eliminar el control de acceso al repositorio.
- Rediseño de las interfaces de usuario.
- Reorganización y modificación de las clases existentes de manera que soporten almacenar el recurso digital.
- Pruebas sobre los cambios realizados.

Fase de Inicio

Análisis del Repositorio de Metadatos para definir los cambios que permitirán almacenar, modificar y eliminar un OA y su metadato.

- Levantamiento de la información.
- Definición de los requerimientos de software basados en la información suministrada por los usuarios y que deben ser satisfechos en el diseño del sistema.
 - El repositorio de metadatos proveerá los servicios para el manejo de OA (almacenamiento, modificación y eliminación) y estos servicios reemplazarán los que actualmente existen en la Capa de servicios de la base de datos de AMBAR.
 - Los cambios en el repositorio de metadatos deberán ser implementados bajo la misma plataforma utilizada para su creación bajo la arquitectura orientada a servicios.
 - La interfaz ofrecida al usuario deberá ser fácil de usar e intuitiva y acoplarse a la interfaz de la Capa de servicios de la base de datos de AMBAR.
 - El repositorio de metadatos, al igual que la capa de servicios de la base de datos de AMBAR, deberá poder manejar formatos de datos de tipo: presentaciones Power Point, aplicaciones java, archivos flash, video, audio, imagen, texto, archivos xml, archivos comprimidos zip.

Fase de Integración

Como ambas aplicaciones poseen control de acceso, es necesario eliminar el control de alguna de las dos. Debido a que la Capa de Servicios contiene funcionalidades adicionales que están disponibles para el usuario y las funciones de Agregar, modificar y eliminar OA van a utilizar los servicios del Repositorio de Metadatos, se decidió eliminar el control de acceso existente en el repositorio de Metadatos y manejar la sesión de usuario a través de la Capa de Servicios.

Debido a que los servicios de agregar y modificar OA, contenidos en la Capa de Servicios de AMBAR, permitían almacenar y descargar, respectivamente, el OA (recurso

digital), se incorporó esta funcionalidad en las modificaciones del Repositorio de Metadata para esas operaciones. Esto es realizado, ya que se debe garantizar que un OA es el resultado de almacenar un recurso digital más su información descriptiva (metadata), esto permite su posterior localización y reutilización.

La eliminación del control de acceso implicó modificar las interfaces de usuario donde se maneja la sesión de usuario. Para las modificaciones descritas anteriormente, se definió generar una instancia de la aplicación metadataWsCliente, llamada “nuevametadata” y así mantener la versión original para casos de errores y pruebas.

Se definieron y diseñaron los diagramas de casos de uso de la aplicación Web que permitirá realizar las pruebas de la capa de Servicios Web de base de datos de AMBAR. Los diagramas de casos de uso representan la forma como un usuario opera con el sistema, además de la forma, tipo y orden en cómo estos elementos interactúan.

Antes de describir los casos de uso, es importante describir los actores que interactúan en el sistema. Un actor se define como el rol o función que asume una persona, sistema o entidad que interactúa con la aplicación que se está desarrollando. Estos actores se describen a continuación:

Actores	Descripción
Usuario general	Este usuario no requiere contraseña para hacer uso del sistema y tiene funcionales restringidas, se correspondería con los que vistan la primera vez la página.
Usuario autorizado	Este usuario puede buscar y bajar OA de la BD de AMBAR, si es autorizado.
Administrador	Este usuario tiene todos los privilegios en el uso del sistema. Se encarga de la administración, actualización de los servicios, el mantenimiento de la BD de AMBAR e interfaz Web, entre otros.

A continuación se muestran los tres niveles de casos de uso definidos, para las funcionalidades de la capa de servicios de la BD del SIG-AMBAR, los casos de uso que fueron modificados se encuentran sombreados en color gris:

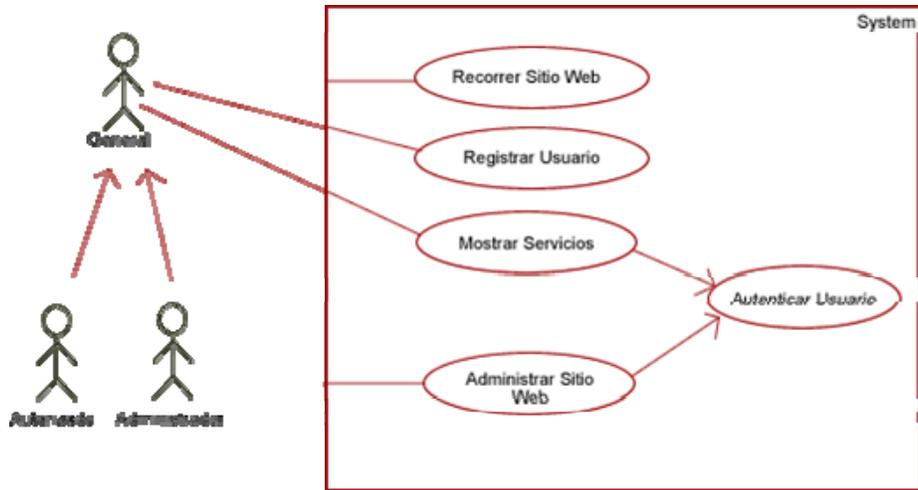


Diagrama de Casos de Uso de interfaz Web - Nivel 1

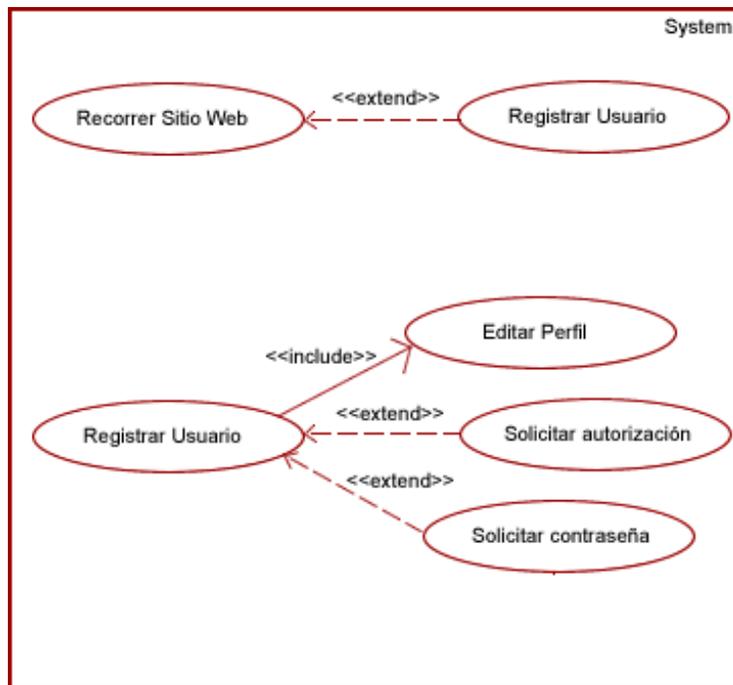


Diagrama de Casos de Uso de interfaz Web - Nivel 2

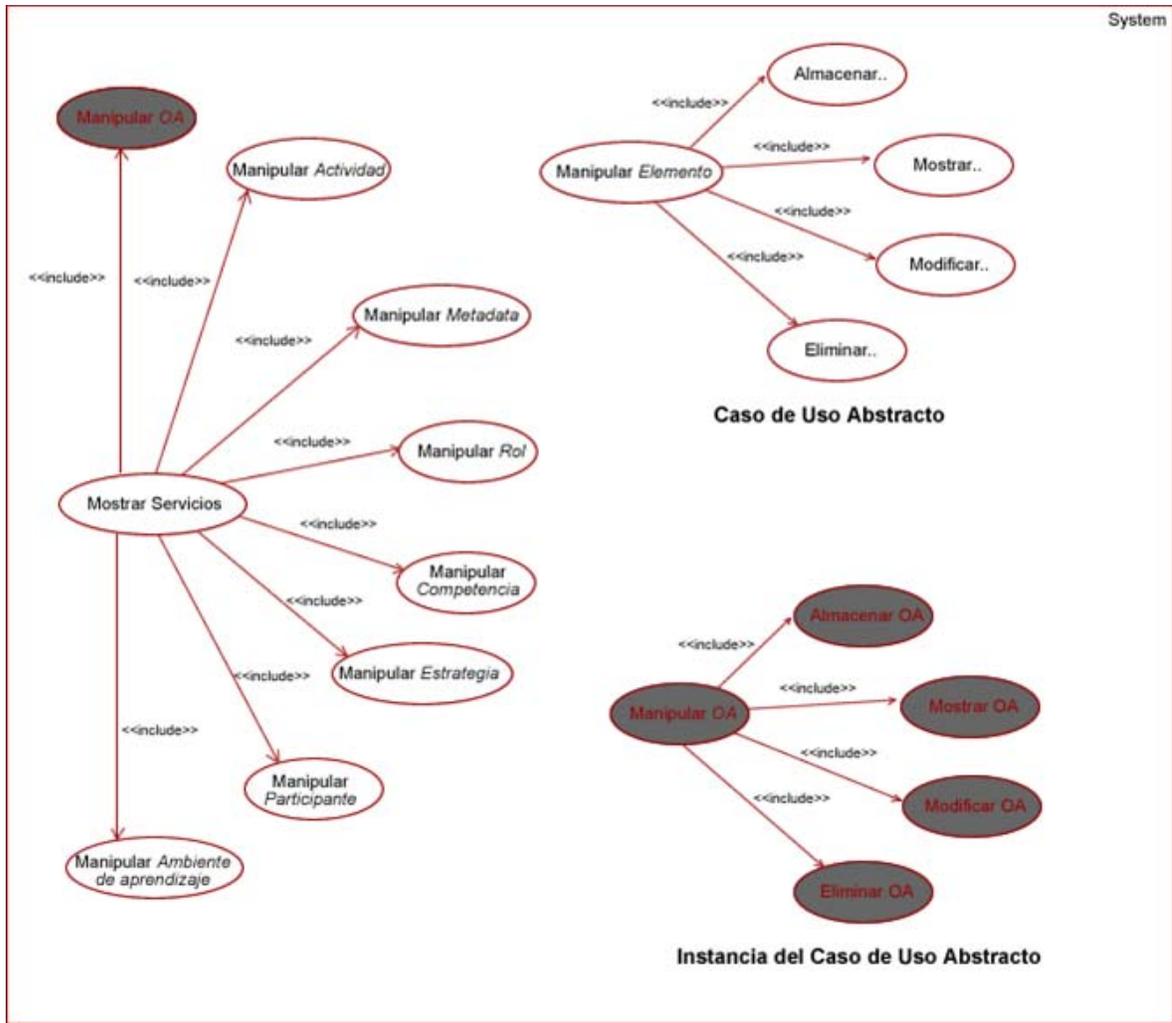


Diagrama de Casos de Uso de interfaz Web - Nivel 3

Descripción de los Casos de Uso

Caso de Uso	Recorrer el sitio Web	
Actores	Usuario general, usuario administrador, usuario autorizado.	
Tipo	Primario	
Descripción	El actor ingresa a la aplicación colocando el URL de la misma en su navegador Web. Aquí los usuarios pueden interactuar con la aplicación Web para conocer en qué consiste el proyecto AMBAR.	
Curso Normal de	Acción De Actor	Acción del Sistema

los Eventos	<p>1. Este caso de uso comienza cuando el actor desea entrar al sitio Web de la aplicación introduciendo en un navegador la dirección URL del sistema.</p> <p>3. El actor recorre el sitio Web sin registro. El actor puede:</p> <ul style="list-style-type: none"> a. Ver una descripción del proyecto AMBAR. b. El actor puede registrarse en el sistema. c. El actor puede solicitar autorización de acceso al sistema. 	<p>2. La aplicación muestra la interfaz de inicio del sitio Web.</p>
Cursos Alternos		
Requerimientos Especiales		
Precondiciones	El actor debe introducir correctamente la URL del sistema para ingresar al sitio Web de la aplicación.	
Post-condiciones	El usuario ingresó a la aplicación Web del sistema.	

Tabla 1. Caso de Uso "Recorrer el Sitio Web".

Caso de Uso	Registrar usuario	
Actores	Usuario general.	
Tipo	Primario	
Descripción	Viene del caso de uso recorrer el sitio Web, aquí el actor puede registrarse en el sistema colocando sus datos en el formulario de registro. Luego que introduce todos los datos solicitados por el sistema y los mismos son válidos, el usuario queda registrado en el sistema con un perfil específico.	
Curso Normal de los Eventos	Acción De Actor	Acción del Sistema
	<p>1. Este caso de uso viene del evento 3 cuando el actor recorre el sitio Web sin registro y desea registrarse en el sistema.</p>	<p>2. La aplicación muestra el formulario</p>

	<p>3. El actor rellena el formulario colocando todos los datos solicitados por el sistema y envía el formulario.</p> <p>5. El actor tiene un usuario y una contraseña válida para ingresar al sistema.</p>	<p>que debe llenar el usuario para realizar su registro en el sistema.</p> <p>4. El sistema verifica que los datos introducidos sean correctos.</p>
Cursos Alternos	6. Si los datos introducidos en el formulario no son válidos, no se permite el registro del usuario en el sistema.	
Requerimientos Especiales		
Precondiciones	El actor debe ingresar al sitio Web.	
Post-condiciones	El usuario fue registrado en el sistema.	

Tabla 2. **Caso de Uso “Registrar Usuario”**

Caso de Uso	<i>Autenticar usuario</i>	
Actores	Usuario general, usuario autorizado.	
Tipo	Primario	
Descripción	El actor intenta ingresar al sistema colocando su usuario y contraseña en los campos de inicio de sesión del componente de seguridad que valida los accesos al sistema, el sistema valida y verifica si el usuario tiene permiso de entrar o no a la aplicación.	
Curso Normal de los Eventos	Acción De Actor	Acción del Sistema
	1. Este caso de uso comienza cuando el actor desea ingresar al sistema colocando su usuario y contraseña de acceso en los campos de inicio de sesión del componente de seguridad.	2. El sistema verifica que los datos introducidos sean correctos y si el usuario

	3. El usuario ingresa al sistema	<p>puede o no ingresar al sistema:</p> <ol style="list-style-type: none"> a. Si el usuario tiene permiso de acceso, ingresa al sistema. b. Si el usuario no tiene permiso de acceso, regresa al paso 1. <p>4. El sistema muestra los servicios disponibles:</p> <ul style="list-style-type: none"> • Agregar en la BD de AMBAR. • Consultar en la BD de AMBAR. • Descargar de la BD de AMBAR.
Cursos Alternos	3. Si el usuario no está registrado o los datos suministrados son incorrectos, no puede ingresar a la aplicación.	
Requerimientos Especiales		
Precondiciones	El usuario debe estar registrado en el sistema.	
Post-condiciones	El usuario fue autenticado y pudo ingresar a la aplicación.	

Tabla 3. Caso de Uso "Autenticar Usuario"

Caso de Uso	<i>Manipular elemento</i>	
Actores	Usuario administrador, usuario autorizado.	
Tipo	Primario	
Descripción	Viene del caso de uso autenticar usuario, aquí el usuario invoca los servicios disponibles de acuerdo a su perfil para interactuar con la BD de AMBAR.	
Curso Normal de	Acción del Actor	Acción del Sistema

los Eventos	<p>1. Este caso viene del evento 5 ingresar al sistema.</p> <p>3. El usuario carga los datos solicitados de acuerdo al servicio seleccionado.</p>	<p>2. De acuerdo al perfil del usuario que ingresa al sistema se le muestran los servicios disponibles:</p> <ul style="list-style-type: none"> 2.1. Consultar en la base de datos AMBAR. 2.2. Almacenar en la BD de AMBAR. 2.3. Descargar de la BD de AMBAR. 2.4. Eliminar de la BD AMBAR. <p>4. El sistema le indica al usuario los pasos a seguir de acuerdo al servicio seleccionado.</p> <p>5. El sistema valida que los datos cargados sean correctos y si es así, ejecuta la opción seleccionada.</p>
Cursos Alternos	6. Si el usuario no carga los datos necesarios correctamente, se le muestra un mensaje indicándole qué datos están incorrectos y que debe hacer en ese caso.	
Requerimientos Especiales		
Precondiciones	Se debe haber iniciado el caso de uso ingresar al sistema.	
Post-condiciones	Se hace uso de los servicios Web para manipular la base de datos de AMBAR.	

Tabla 4. Caso de Uso "Manipular Elemento"

Caso de Uso	Almacenar OA
Actores	Usuario administrador, usuario autorizado.
Tipo	Primario
Descripción	Viene del Caso de Uso Manipular elemento, aquí el actor

	Agrega un OA y su metadata en la BD.	
Curso normal de los Eventos	Acción de Actor	Acción del Sistema
	<ol style="list-style-type: none"> 1. Este Caso de Uso viene del evento 2.2 de manipular elemento. 2. El actor carga el OA y los datos solicitados para Agregar su información descriptiva. 	<ol style="list-style-type: none"> 3. El sistema le indica los pasos a seguir para introducir la Metadata. 4. El sistema verifica que los datos introducidos sean correctos, de ser así ejecuta la acción seleccionada.
Cursos Alternos	5. Si el actor no ingresa los datos correctamente, se muestra un mensaje de error y se indica la acción a tomar.	
Requerimientos Especiales		
Precondiciones	Se debe haber iniciado el caso de uso Manipular elemento.	
Post-condiciones	Se hace uso de los Web Services para Agregar Metadata a la base de datos.	

Tabla 5. **Caso de Uso “Almacenar OA”**

Caso de Uso	Mostrar OA	
Actores	Usuario administrador, usuario autorizado.	
Tipo	Primario	
Descripción	Viene del evento 2.3 del Caso de Uso Manipular Elemento, aquí el actor Visualiza un OA y su Metadata.	
Curso normal de los Eventos	Acción de Actor	Acción del Sistema
	<ol style="list-style-type: none"> 1. Este Caso de Uso comienza cuando el actor desea visualizar un OA y Metadata. 3. El actor selecciona el tipo 	<ol style="list-style-type: none"> 2. La aplicación muestra las opciones de búsqueda.

	<p>de búsqueda e ingresa un parámetro.</p> <p>5. El actor selecciona la Metadata a visualizar.</p> <p>7. El actor visualiza la Metadata y si desea selecciona el enlace para ver el OA.</p>	<p>4. El sistema muestra los resultados de la búsqueda según los parámetros introducidos por el actor.</p> <p>6. El sistema muestra la Metadata seleccionada e la opción de ver el OA.</p> <p>8. El sistema muestra el OA.</p>
Cursos Alternos	9. Si el actor no ingresa los datos correctamente, se muestra un mensaje de error y se indica la acción a tomar.	
Requerimientos Especiales		
Precondiciones	Se debe haber iniciado el caso de uso Manipular elemento.	
Post-condiciones	Se hace uso de los WebServices para Modificar Metadata.	

Tabla 6. Caso de Uso "Mostrar OA"

Caso de Uso	Modificar OA	
Actores	Usuario administrador, usuario autorizado.	
Tipo	Primario	
Descripción	Viene del evento 2.3 del Caso de Uso Manipular Elemento, aquí el actor Modifica Metadata de un OA en la BD.	
Curso normal de los Eventos	Acción de Actor	Acción del Sistema
	<p>2. Este Caso de Uso comienza cuando el actor desea modificar Metadata.</p> <p>4. El actor selecciona el tipo de búsqueda e ingresa un</p>	<p>3. La aplicación muestra las opciones de búsqueda.</p>

	<p>parámetro.</p> <p>6. El actor selecciona la Metadata a modificar.</p> <p>10. El actor modifica la Metadata.</p>	<p>5. El sistema muestra los resultados de la búsqueda según los parámetros introducidos por el actor.</p> <p>7. El sistema muestra la Metadata seleccionada e indica los pasos a seguir para modificar la misma.</p> <p>11. El sistema verifica que los datos introducidos sean correctos, de ser así ejecuta la acción seleccionada.</p>
Cursos Alternos	12. Si el actor no ingresa los datos correctamente, se muestra un mensaje de error y se indica la acción a tomar.	
Requerimientos Especiales		
Precondiciones	Se debe haber iniciado el caso de uso Manipular elemento.	
Post-condiciones	Se hace uso de los WebServices para Modificar Metadata.	

Tabla 7. Caso de Uso "Modificar OA"

Caso de Uso	<i>Eliminar OA</i>	
Actores	Usuario administrador, usuario autorizado.	
Tipo	Primario	
Descripción	Viene del evento 2.4 del Caso de Uso Manipular elemento, aquí el usuario Elimina un OA y su Metadata de la BD.	
Curso normal de los Eventos	Acción de Actor	Acción del Sistema
	1. Este Caso de Uso comienza cuando el usuario desea eliminar un OA y su metadata.	2. La aplicación muestra las opciones de búsqueda.
	3. El usuario selecciona el	

	<p>tipo de búsqueda e ingresa un parámetro.</p> <p>5. El usuario selecciona el OA a eliminar.</p> <p>13. El usuario elimina la Metadata.</p>	<p>4. El sistema muestra los resultados de la búsqueda según los parámetros introducidos por el usuario.</p> <p>8. El sistema muestra el OA seleccionado e indica los pasos a seguir para eliminar la Metadata.</p> <p>14. El sistema ejecuta la acción seleccionada.</p>
Cursos Alternos	15. Si el usuario no ingresa los datos correctamente, se muestra un mensaje de error y se indica la acción a tomar.	
Requerimientos Especiales		
Precondiciones	Se debe haber iniciado el caso de uso Manipular elemento.	
Post-condiciones	Se hace uso de los Web Services para Eliminar Metadata.	

Tabla 8. **Caso de Uso “Eliminar OA”**

Caso de Uso	Administrar sitio Web	
Actores	Usuario Administrador	
Tipo	Primario	
Descripción	El administrador tiene acceso a todas las opciones del sistema y es el encargado actualizar y mantener toda la información de la aplicación.	
Curso Normal de los Eventos	Acción del Actor	Acción del Sistema
	1. Este Caso de uso se inicia cuando el actor acceder al sistema con privilegios de administrador.	2. El sistema le muestra todos los servicios disponibles para acceder a la BD de AMBAR. También le permite modificar la información de las diferentes capas que conforman

	<p>3. El administrador selecciona la opción que desea modificar.</p> <p>5. El Administrador selecciona la opción que desee actualizar.</p> <p>7. El administrador suministra los datos necesarios para la opción seleccionada.</p>	<p>la arquitectura del sistema (lógica, presentación y acceso a los datos a través de los Servicios Web)</p> <p>4. El sistema le indica al actor las opciones para manipular la BD de AMBAR:</p> <p>4.1. Almacenar en la BD AMBAR. 4.2. Consultar en la BD AMBAR. 4.3. Descargar de la BD AMBAR. 4.4. Eliminar de la BD AMBAR.</p> <p>6. El sistema muestra los formularios correspondientes a la opción:</p> <p>a. Almacenar en la BD AMBAR. b. Consultar en la BD AMBAR. c. Descargar de la BD AMBAR. d. Eliminar de la BD AMBAR.</p> <p>8. Los cambios realizados se almacenan.</p>
Cursos Alternos		
Requerimientos Especiales		
Precondiciones	El usuario debe estar autorizado como administrador del sistema para poder acceder a toda la información de la aplicación.	
Post-condiciones	Todas las actualizaciones realizadas por el administrador se ven reflejadas en la aplicación como mantenimiento del sistema.	

Tabla 9. Caso de Uso "Administrar Sitio Web"

Luego de describir los casos de uso, se definió la arquitectura de la solución de la Integración de la BD del SIG-AMBAR y el Repositorio de Metadata.



Figura 9. Arquitectura de la Integración de la BD del SIG-AMBAR y el Repositorio de Metadata

El usuario interactúa con el repositorio a través de una interface WEB. Esta interface está desarrollada con el lenguaje de programación JSP, el cual se encarga de invocar los servicios web que abstraen la comunicación con las BD. Estos clientes de servicios WEB reciben parámetros que son usados para su procesamiento interno y toma decisiones. Luego, envían sobres de solicitud de información para consumir los servicios WEB.

A través de NetBeans IDE, apoyado en la plataforma Axis2, teniendo los proyectos originales del repositorio de metadata cargado, se creó un nuevo proyecto llamado *nuevametadata* será el nuevo cliente que sustituirá a la anterior aplicación *metadataWsCliente* del Repositorio de Metadata, como se describe a continuación:

- Se ubica el proyecto desarrollado en NetBeans y haciendo clic derecho sobre él, se selecciona **New → Web Service...**

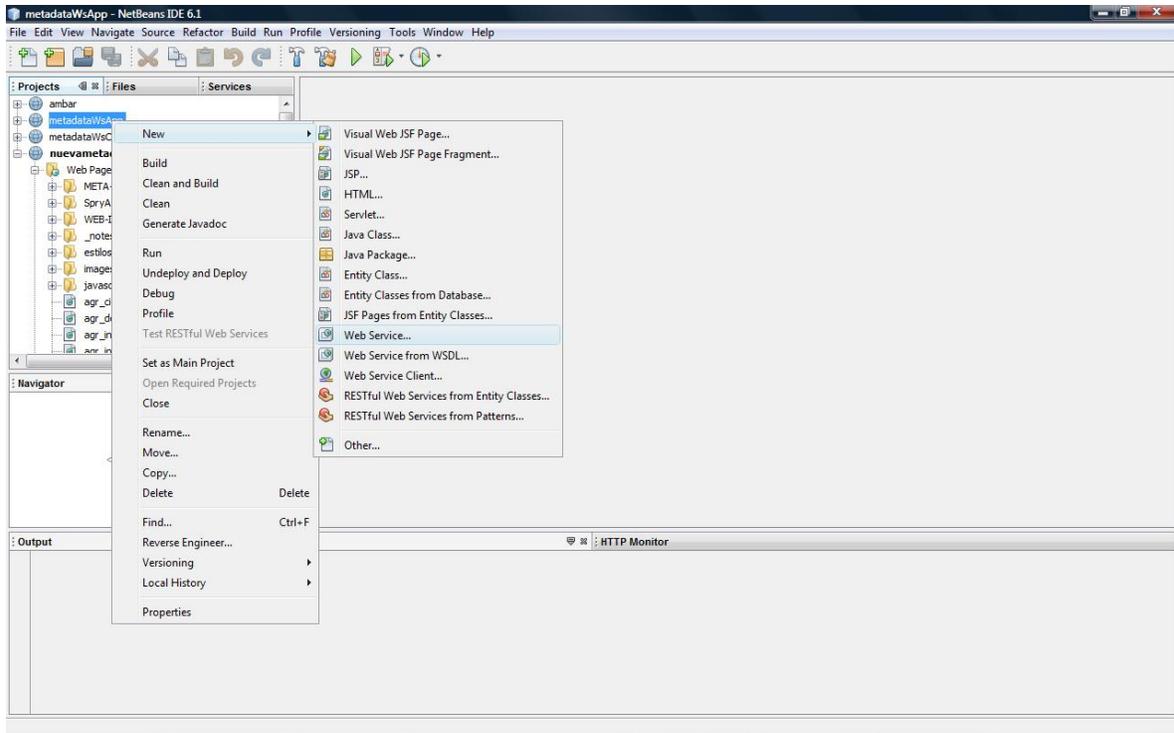


Figura 10. Creación de un servicio web

- Se coloca el nombre del nuevo Servicio Web, para este caso, llamado nuevamedata y se indica la carpeta de trabajo, que en este trabajo es llamada tesis2.

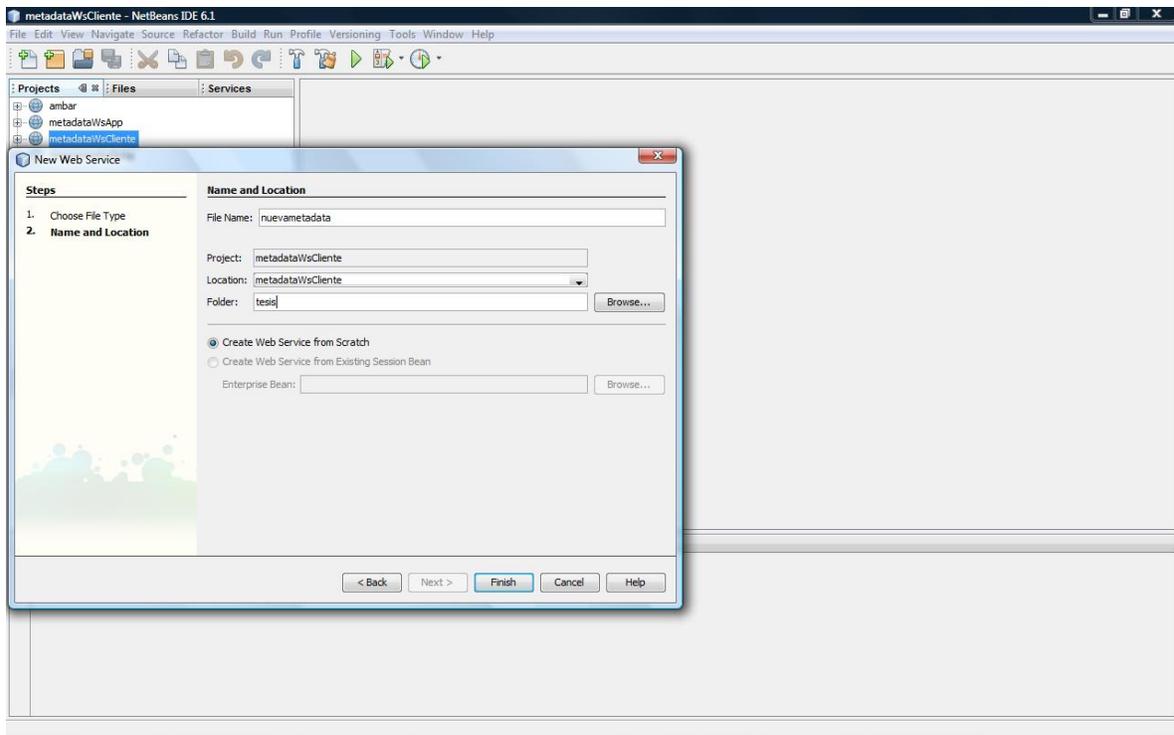


Figura 11. Ingreso de nombre y ubicación del Servicio Web

Las interfaces correspondientes a los métodos de Agregar, Modificar, eliminar metadata son reorganizadas y rediseñadas. Inicialmente, todas las clases se manejaban en un mismo .html o .jsp, por lo que era más tediosa su modificación. En base a eso, por cada categoría del estándar LOM, se creó un archivo .html o .jsp dependiendo si poseían contenido dinámico o no. Esto fue realizado para proporcionar futuras facilidades de mantenimiento o nuevas versiones a la aplicación.

El control de acceso existente fue eliminado de manera que la sesión de usuario se maneje desde la Capa de Servicios de la base de datos de AMBAR.

Control de Acceso

```
<%
if (session.getAttribute("rol") == null || session.getAttribute("rol").equals("No autorizado"))
    response.sendRedirect("index.jsp");
%>
```

El Servicio Web cliente *nuevametadata*, interactúa con el repositorio a través de las interfaces WEB descritas anteriormente. Estas interfaces se encargan de invocar los

servicios web que abstraen la comunicación con la BD y luego envían la solicitud de información usando el protocolo SOAP para consumir los servicios web.

Los servicios WEB al recibir la petición y comunicarse con la BD, procesan el requerimiento y luego envían la respuesta. Cuando el cliente de servicio WEB recibe la respuesta, se encarga de devolver los datos a la página JSP, para que esta los trabaje. En este punto, se incorporó la funcionalidad de solicitar el recurso digital del OA, para ello se crearon 4 archivos que permitieron el tratamiento del objeto para que este sea almacenado en la BD del SIG-AMBAR. Estos archivos son los siguientes:

- **New_upload.jsp**

Este archivo permite definir donde va a ser guardado el OA al utilizar el servicio correspondiente a “Almacenar OA y su información descriptiva”.

- **New_saveFile.jsp**

Este archivo es invocado al momento de almacenar el OA para que este haga uso del servicio correspondiente al almacenamiento del mismo.

- **New_saveName.jsp**

Este archivo permite guardar la identificación del OA y su extensión en un archivo .properties que se crea automáticamente.

- **New_getextension.jsp**

Este archivo permite recuperar la extensión de un objeto cuando se realiza una búsqueda.

Los 4 archivos descritos anteriormente fueron creados y ubicados en la carpeta “aplicación_ambar” correspondiente a la Capa de Servicios Web de la BD del SIG-

AMBAR, esto con la finalidad de permitir el uso de los servicios web asociados ya creados para la manipulación de OA en la BD del SIG-AMBAR.

Adicionalmente, se creó otra interfaz que permitiera almacenar solo información descriptiva del OA, para que desde la Capa de Servicios se pudiera manipular información descriptiva de un OA que pudiera estar almacenado en otro Repositorio.

Fase de Transición

En esta fase se realizaron las siguientes tareas:

- Se completó la documentación requerida por los usuarios.
- Se realizaron pruebas sobre los cambios realizados. Debido a que la página inicial y el control de acceso fueron eliminados, se creó una página sencilla, que luego fue eliminada, que permitió realizar las pruebas sobre los servicios disponibles y así simular como sería el comportamiento del repositorio de metadata cuando se integrara con la Capa de Servicios. La página de prueba contiene los enlaces a los siguientes servicios:
 - Agregar metadata
 - Modificar metadata
 - Eliminar metadata

Una vez completadas las pruebas satisfactoriamente, se procede a la segunda iteración de este trabajo.

4.2. Segunda Iteración

Fase de Inicio

En esta fase se realizó la siguiente tarea:

- Durante esta iteración, se acopló el producto funcional obtenido de la primera iteración y se realizan las adaptaciones de las nuevas funcionalidades solicitadas por el usuario. Se definió el Puente de conexión entre la Capa de Servicios de la BD de AMBAR y el Repositorio de Metadata.

- Se define la modificación en el modelo conceptual de la BD AMBAR y sus interfaces.

Fase de Integración

En base a los requerimientos definidos por los usuarios se contemplan los siguientes ajustes sobre la Capa de Servicios:

- Modificación del Modelo conceptual actual de la BD del SIG-AMBAR, de manera que sea posible su integración con el repositorio de Metadata. Cada OA almacenado, modificado o eliminado de la BD manejará su metadata a través del Repositorio de metadata y no por medio de la clase Metadata local. Es importante resaltar, que la clase Metadata no es eliminada del modelo debido a que otras clases hacen uso de ella, estas clases son Actividad y Ambiente de Aprendizaje.
- Se realizó una revisión de las interfaces de usuario, eliminando así, enlaces sin uso e información desactualizada y por petición de los usuarios se eliminaron los servicios correspondientes a Prerequisitos y Servicios.

A continuación se ilustra el modelo conceptual de la integración:

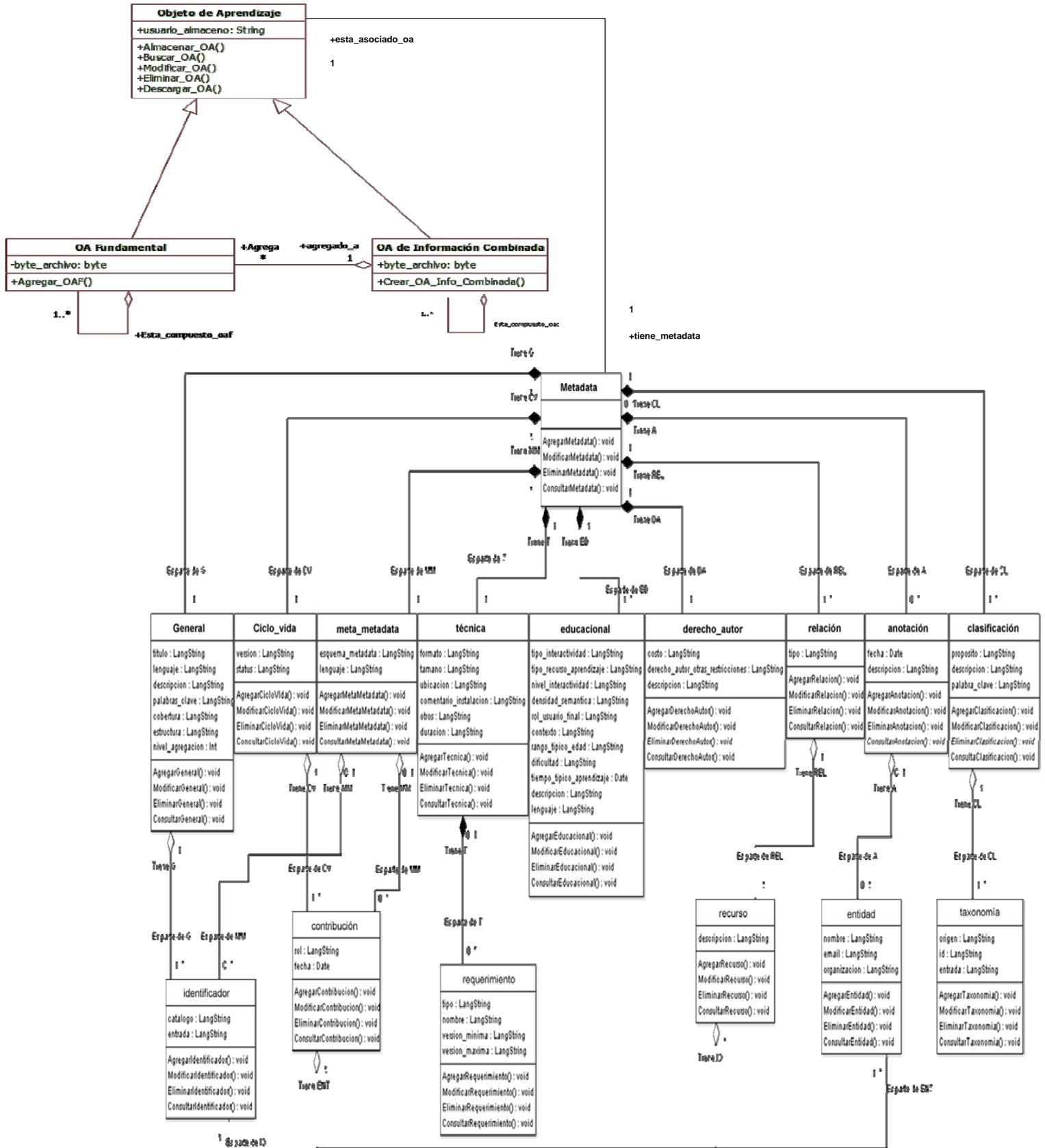


Figura 12. Modelo Conceptual de la integración

En este modelo se resaltan sólo las clases asociadas a Objeto de Aprendizaje ya que fueron las que se integraron con el repositorio de metadata.

- Se creó un Servlet que permite que el Repositorio de Metadata pueda manejar la misma sesión de usuario que la Capa de Servicios, incorporándose un control de acceso único para ambos proyectos y así permitir que estos dos sean vistos como un proyecto único y garantizando que el repositorio de Metadata no tenga acceso independiente.

El servlet es llamado *bridgeright.java* y se describe a continuación:

```
import java.io.*;
import java.net.*;

import javax.servlet.*;
import javax.servlet.http.*;

/**
 *
 * @author joseline
 */
public class Bridgeright extends HttpServlet {

    /**
     * Handles the HTTP <code>GET</code> method.
     * @param request servlet request
     * @param response servlet response
     */
    protected void doGet(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException {
    }

    /**
     * Handles the HTTP <code>POST</code> method.
     * @param request servlet request
     * @param response servlet response
     */
    protected void doPost(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException {
        HttpSession session = request.getSession();
        Object aux = session.getValue("Activa");
        String estado = String.valueOf(aux);
        Object usuario = session.getValue("usuario");//Se obtiene el atributo usuario del participante
        String log = String.valueOf(usuario);
        String pagina = request.getParameter("pagina");
        if ((session.getValue("usuario") == null) && (estado != "activa")) {
            if ((pagina != null) && (!pagina.equals("4"))) {
                session.putValue("Activa", request.getParameter("estado"));
                session.putValue("usuario", request.getParameter("log"));
            }
        }
        if (pagina == null) {
            response.sendRedirect("http://localhost:8080/aplicacion_ambar/");
        }
        else {
            if (pagina.equals("1")) {
                response.sendRedirect("http:// 150.185.75.79:8080/nuevametadada/agregar_metadata.jsp");
            }
        }
    }
}
```

```

        else if (pagina.equals("2")) {
            response.sendRedirect("http://
150.185.75.79:8080/nuevametadada/buscar_metadata.jsp?proposito=modificar");
        }
        else if (pagina.equals("3")) {
            response.sendRedirect("http://
150.185.75.79:8080/nuevametadada/buscar_metadata.jsp?proposito=eliminar");
        }
        else if (pagina.equals("4")) {
            response.sendRedirect("http://
150.185.75.79:8080/nuevametadada/buscar_metadata.jsp?proposito=buscar");
        }
        else if (pagina.equals("5")) {
            response.sendRedirect("http://150.185.75.79:8080/nuevametadada/agregar_metadata1.jsp");
        }
    }
    else {
        response.sendRedirect("http://localhost:8080/aplicacion_ambar/");
    }
}

/**
 * Returns a short description of the servlet.
 */
public String getServletInfo() {
    return "Short description";
}
// </editor-fold>
}

```

Los datos de sesión son recibidos, donde lo primero que se valida es que sea un usuario válido y que la sesión aún se encuentra activa desde la Capa de Servicios Web. Si la sesión a expirado, automáticamente se muestra la página con el mensaje correspondiente a Sesión finalizada y el usuario tendrá que iniciar sesión nuevamente desde la página principal. Este procedimiento garantiza que no se pueda acceder al repositorio de metadata sin que sea un usuario autorizado y si el usuario ya había finalizado su sesión.

Es importante resaltar que para que el servlet pueda funcionar, se modificó el archivo *web.xml*, ubicado en la carpeta de trabajo (...NetBeansProjects\nuevametadada\web\WEB-INF) del Servicio Web creado en la iteración anterior (*nuevametadada*). En este archivo se agregó una nueva entrada que permite registrar el nuevo servlet creado:

```

<?xml version="1.0" encoding="UTF-8" ?>
<web-app version="2.5" xmlns="http://java.sun.com/xml/ns/javaee" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xsi:schemaLocation="http://java.sun.com/xml/ns/javaee http://java.sun.com/xml/ns/javaee/web-
app_2_5.xsd">
<servlet>
<servlet-name>wsClienteServlet</servlet-name>
<servlet-class>tesis.cliente.wsClienteServlet</servlet-class>
</servlet>
</servlet>

```

```

<servlet-name>wsServletBuscar</servlet-name>
<servlet-class>tesis.cliente.wsServletBuscar</servlet-class>
</servlet>
= <servlet>
  <servlet-name>autenticar</servlet-name>
  <servlet-class>tesis.cliente.autenticar</servlet-class>
</servlet>
= <servlet>
  <servlet-name>servletAgregarMetadata</servlet-name>
  <servlet-class>tesis.cliente.servletAgregarMetadata</servlet-class>
</servlet>
= <servlet>
  <servlet-name>servletBusquedaBasica</servlet-name>
  <servlet-class>tesis.cliente.servletBusquedaBasica</servlet-class>
</servlet>
= <servlet>
  <servlet-name>wsServletEliminarPorOID</servlet-name>
  <servlet-class>tesis.cliente.wsServletEliminarPorOID</servlet-class>
</servlet>
= <servlet>
  <servlet-name>wsServletBuscarPorOID</servlet-name>
  <servlet-class>tesis.cliente.wsServletBuscarPorOID</servlet-class>
</servlet>
= <servlet>
  <servlet-name>wsServletModificarPorOID</servlet-name>
  <servlet-class>tesis.cliente.wsServletModificarPorOID</servlet-class>
</servlet>
= <servlet>
  <servlet-name>wsServletEliminarPorOIDsilencioso</servlet-name>
  <servlet-class>tesis.cliente.wsServletEliminarPorOIDsilencioso</servlet-class>
</servlet>
= <servlet>
  <servlet-name>wsServletBusquedaAvanzada</servlet-name>
  <servlet-class>tesis.cliente.wsServletBusquedaAvanzada</servlet-class>
</servlet>
= <servlet>
  <servlet-name>wsServletBuscarPorOidXML</servlet-name>
  <servlet-class>tesis.cliente.wsServletBuscarPorOidXML</servlet-class>
</servlet>
= <servlet>
  <servlet-name>Bridgeright</servlet-name>
  <servlet-class>Bridgeright</servlet-class>
</servlet>
= <servlet-mapping>
  <servlet-name>wsClienteServlet</servlet-name>
  <url-pattern>/wsClienteServlet</url-pattern>
</servlet-mapping>
= <servlet-mapping>
  <servlet-name>wsServletBuscar</servlet-name>
  <url-pattern>/wsServletBuscar</url-pattern>
</servlet-mapping>
= <servlet-mapping>
  <servlet-name>autenticar</servlet-name>
  <url-pattern>/autenticar</url-pattern>
</servlet-mapping>
= <servlet-mapping>
  <servlet-name>servletAgregarMetadata</servlet-name>
  <url-pattern>/servletAgregarMetadata</url-pattern>
</servlet-mapping>
= <servlet-mapping>
  <servlet-name>servletBusquedaBasica</servlet-name>
  <url-pattern>/servletBusquedaBasica</url-pattern>
</servlet-mapping>
= <servlet-mapping>
  <servlet-name>wsServletEliminarPorOID</servlet-name>
  <url-pattern>/wsServletEliminarPorOID</url-pattern>
</servlet-mapping>
= <servlet-mapping>
  <servlet-name>wsServletBuscarPorOID</servlet-name>
  <url-pattern>/wsServletBuscarPorOID</url-pattern>

```



Registro del Servlet Bridgeright

```

</servlet-mapping>
- <servlet-mapping>
  <servlet-name>wsServletModificarPorOID</servlet-name>
  <url-pattern>/wsServletModificarPorOID</url-pattern>
</servlet-mapping>
- <servlet-mapping>
  <servlet-name>wsServletEliminarPorOIDsilencioso</servlet-name>
  <url-pattern>/wsServletEliminarPorOIDsilencioso</url-pattern>
</servlet-mapping>
- <servlet-mapping>
  <servlet-name>wsServletBusquedaAvanzada</servlet-name>
  <url-pattern>/wsServletBusquedaAvanzada</url-pattern>
</servlet-mapping>
- <servlet-mapping>
  <servlet-name>wsServletBuscarPorOidXML</servlet-name>
  <url-pattern>/wsServletBuscarPorOidXML</url-pattern>
</servlet-mapping>
- <servlet-mapping>
  <servlet-name>Bridgeright</servlet-name>
  <url-pattern>/Bridgeright</url-pattern>
</servlet-mapping>
- <session-config>
  <session-timeout>30</session-timeout>
</session-config>
- <welcome-file-list>
  <welcome-file>index.jsp</welcome-file>
</welcome-file-list>
</web-app>

```

} Registro del Servlet Bridgeright

Para que el servlet, descrito anteriormente, obtenga los datos correspondientes a la información de la sesión, se creó un archivo JSP (bridgeleft.jsp) que es invocado al hacer clic en los servicios de agregar, modificar o eliminar objeto de aprendizaje desde la Capa de Servicios.

Bridgeleft.jsp verifica que el usuario sea del sistema obteniendo el atributo del usuario participante e invoca al servlet *bridgeright.java* por post con sus parámetros.

A continuación se muestra el código para el bridgeleft.jsp:

```

<%
    //Tratamiento de la sesion
    if (session == null) {
        response.sendRedirect("pagina_principal.jsp");
    }
    Object aux = session.getValue("Activa");
    String estado = String.valueOf(aux);
    Object usuario = session.getValue("usuario");//Se obtiene el atributo usuario del participante
    String log = String.valueOf(usuario);
    String pagina = request.getParameter("pagina");

```

```

        if ((session.getValue("usuario") == null) && (estado != "activa") && ((pagina ==
null)||(!pagina.equals("4")))) {
            //Se verifica que el usuario sea del sistema y no un intruso
            response.sendRedirect("pagina_fin_sesion.jsp");
        }
        else {
%>
<html>
<body>
<form name="form1" method="post" action="http:// 150.185.75.79:8080/nuevametadata/Bridgeright">
    <input type="hidden" name="estado" value="<%=estado%>" />
    <input type="hidden" name="log" value="<%=log%>" />
    <input type="hidden" name="pagina" value="<%=pagina%>" />
</form>
<script type="text/javascript" language="javascript">
    document.form1.submit();
</script>
</body>
</html>
<%
        //Fin del else tratamiento de la sesion
%>

```

Una vez enviados los parámetros al servlet, este siempre va a ejecutar dos validaciones:

1. Si la sesión esta activa, este direcciona automáticamente a la página (del repositorio de metadata) que se estaba invocando.
2. Si la sesión no está activa del lado del Repositorio de Metadata, se toman los parámetros y se crea la misma sesión obtenida en la Capa de Servicios y luego de creada, se direcciona a la página que se estaba invocando.

Si es la primera vez que se intenta acceder al Repositorio de Metadata, habiendo un usuario iniciado sesión en la Capa de Servicios, la sesión nunca estará activa y por tanto, deberá ser creada.

Luego de creados los archivos descritos anteriormente, se procedió a modificar todas las interfaces de usuario que hacen llamado a agregar, modificar y eliminar objetos de aprendizaje (Ejemplo: `pagina_administrador.jsp`), para direccionarlos al archivo `bridgeleft.jsp` y se realicen las validaciones correspondientes.

Luego, se modificaron de nuevo las clases del Servicio Web nuevametadata, con el fin de incorporarles el control de acceso. El código que se le agrego fue el siguiente:

```
<%
//Tratamiento de la sesion
if (session == null) {
    response.sendRedirect("http://150.175.75.79:8080/aplicacion_ambar/");
}
Object aux = session.getValue("Activa");
String estado = String.valueOf(aux);
Object usuario = session.getValue("usuario");//Se obtiene el atributo usuario del participante
String log = String.valueOf(usuario);
if ((session.getValue("usuario") == null) && (estado != "activa")) {
    //Se verifica que el usuario sea del sistema y no un intruso
    response.sendRedirect("http:// 150.175.75.79:8080/aplicacion_ambar/pagina_fin_sesion.jsp");
}
else {
%>
```

Luego de tener el modelo conceptual final, se procedió a rediseñar todas las interfaces de usuario de manera que se obtuviera una interfaz definitiva ajustada con los nuevos servicios incorporados en la iteración 1.

Una vez realizados todos los cambios planteados, se mostrará la vista final de la interfaz de usuario, obteniendo así un prototipo de aplicación que se mantendrá en revisión para seguir ajustándolo a los próximos proyectos a implementar sobre AMBAR.

Se describirán solo las actividades de agregar, modificar y eliminar objetos de aprendizaje, los servicios restantes mostrados en la Capa de servicios, funcionalmente se mantienen igual, solo se rediseñó la interfaz correspondiente.

En la Figura 13, se puede observar la página principal que interactúa con la Capa de Servicios de la base de datos de AMBAR, donde para ingresar a la aplicación debe ser un usuario registrado o administrador. La dirección donde se encuentra la aplicación es **http://150.185.75.79:8080/aplicacion_ambar/pagina_principal.jsp**



Figura 13. Pantalla inicial - Capa de Servicios

En la Figura 14, se muestra el proceso de autenticación si es usuario administrador. Luego de ingresar el usuario y contraseña, debe hacer clic sobre el botón Ingresar para acceder a las funciones permitidas para ese usuario.



Figura 14. Autenticación – Usuario Administrador

En la Figura 15, una vez ingresado como Administrador podrá visualizar todos los servicios que este usuario tiene disponibles.

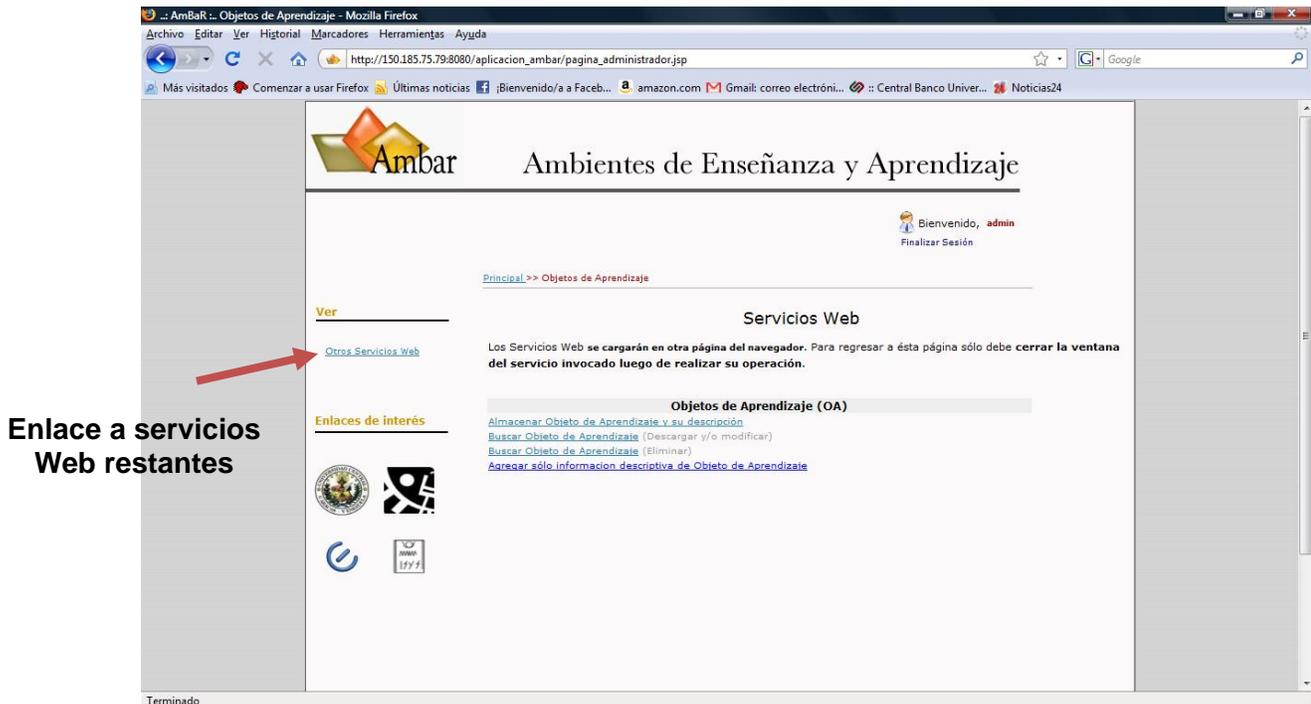


Figura 15. Página principal Administrador

Como servicios principales se dejaron sólo los de OA, debido que a son los que están integrados con el repositorio de Metadata, los servicios restantes se colocaron en un enlace a parte desde la página principal del administrador. La página se puede observar en la siguiente figura:



Figura 16. Página administrador – Otros Servicios Web

En la Figura 17, si hace clic sobre el enlace “*Almacenar Objeto de Aprendizaje y su descripción*” desde la página principal del administrador (Ver figura 15), se muestra el formulario para Agregar un objeto de Aprendizaje junto con su información descriptiva.



Figura 17. Agregar OA y su información descriptiva

Se debe seleccionar el recurso digital, presionando el botón Examinar y seguidamente llenar la información que describe al objeto. Luego de llenar todos los datos y presionar el botón “Agregar”, se indicara que la operación fue realizada satisfactoriamente, como se muestra en la siguiente figura:



Figura 18. Agregar OA – Operación exitosa

Si desea buscar un OA ya almacenado para descargarlo el recurso digital o modificar la información descriptiva del mismo, desde la página donde se listan los servicios Web debe hacer clic en el enlace a Buscar Objeto de Aprendizaje (Modificar y/o descargar) para buscar el OA (Ver Figura 15).

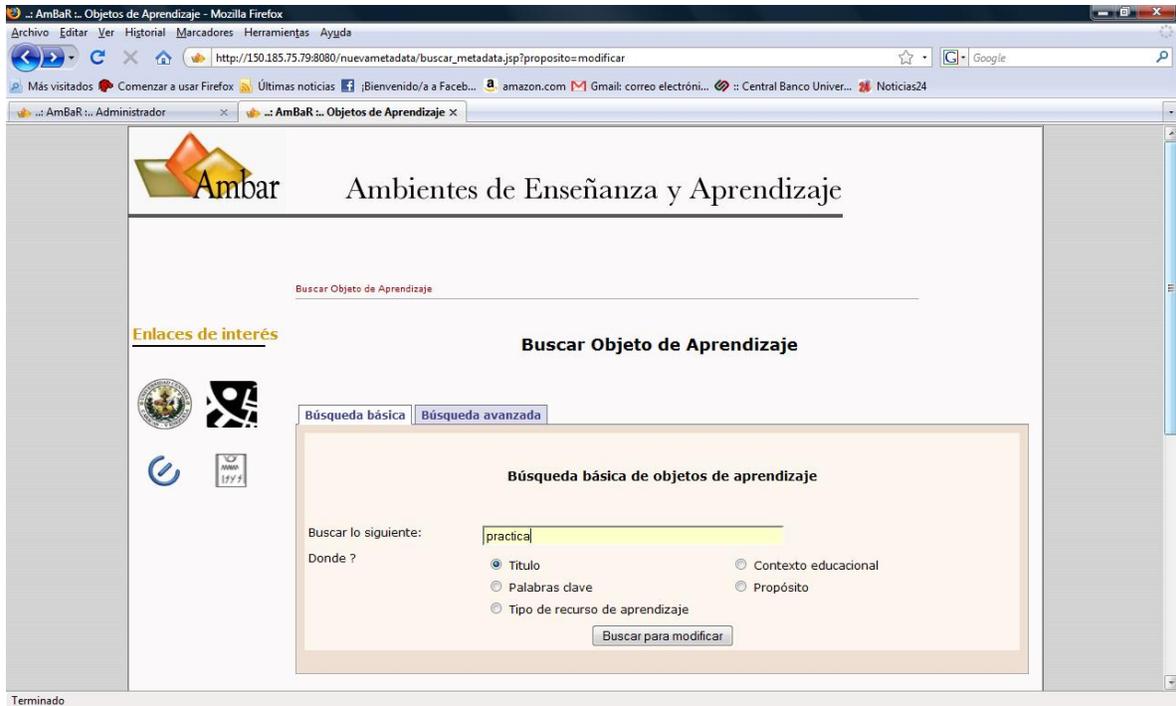


Figura 19. Búsqueda de OA - Modificar

En la Figura 20, se visualiza el resultado de una búsqueda



Figura 20. Resultado de búsqueda de OA a modificar

Si hace clic sobre la palabra “Modificar” podrá visualizar, en modo edición, la descripción del OA.

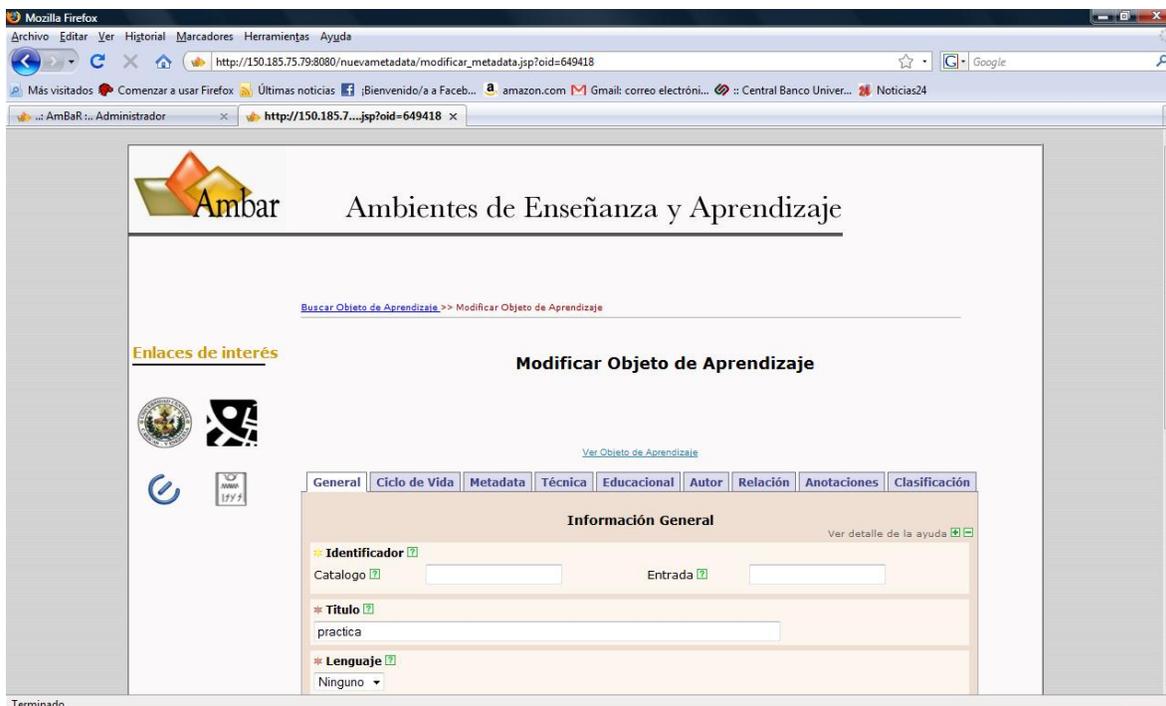


Figura 21. Modificar OA

Si desea ver el recurso digital, debe hacer clic sobre el enlace “Ver Objeto de Aprendizaje” y se desplegará el OA para luego descargarlo si así se requiere. En la Figura 22, se muestra el recurso digital asociado a la descripción mostrada.



Figura 22. Ver recurso digital

Fase de Transición

En esta fase se realizó lo siguiente:

- Se completó toda la documentación requerida por el usuario para la iteración en curso, correspondiente al Trabajo Especial de Grado.
- Se entregó una versión funcional de los sistemas integrados.
- Se realizaron los cambios y modificaciones solicitados por el usuario.

CONCLUSIONES

Una vez concluido este trabajo especial de grado, se cumple con el objetivo planteado con la finalidad de integrar la base de datos del SIG-AMBAR con el Repositorio de Metadata. Permitiendo así, darle una mejor utilidad a ambas aplicaciones que inicialmente eran independientes.

El método de desarrollo empleado fue UP Ágil, debido a que constituye el método estándar más utilizado para el análisis, implementación y documentación de sistemas orientados a objetos, teniendo como primordial característica la adaptabilidad para dar respuesta eficaz en cada una de sus fases de desarrollo.

La arquitectura orientada a servicios empleada inicialmente para la implementación de la BD del SIG-AMBAR y el repositorio de metadata, facilitó el proceso de integración y adaptación entre ambas aplicaciones, permitiéndose así el uso de los servicios ya existentes y la incorporación de elementos que soportaran el uso de esos servicios.

La importancia de la integración realizada en este trabajo, radica en que ahora el proyecto AMBAR contará con un sistema mejorado que permite tener más organizada la información de los OA con una estructura estandarizada de almacenamiento y, que a su vez, proporciona mecanismos eficientes de catalogación, descripción y localización de los mismos.

RECOMENDACIONES

Una vez concluido el Trabajo Especial de Grado, y establecidas las conclusiones del mismo, surgen las siguientes recomendaciones:

- Incorporar en el Repositorio de Metadata, servicios de búsquedas inteligentes que se encarguen de aplicar algoritmos de búsquedas con mayor semántica en base al contexto que el usuario defina o algún comportamiento particular.
- Implementar el estándar IMS Digital Repositories Interoperability (DRI), el cual está orientado específicamente a los repositorios de OAs y a facilitar la interacción entre ellos, mediante la estandarización de las operaciones más comunes. La especificación facilita un esquema funcional de la arquitectura del sistema y un modelo de referencia completo para la interoperabilidad de repositorios.

REFERENCIAS BIBLIOGRÁFICAS

- ADL. (2007). *Advanced Distributed Learning*. ADL Releases SCORM 2004 3rd Edition Content Examples Updates. Recuperado el 20 de abril de 2007, de <http://www.adlnet.gov/>.
- Aguilar, C., Zechinelli Martín, J. L., Muñoz Arteaga, J., 2003, "[Hacia la creación y administración de repositorios de objetos de aprendizaje](#)". Disponible en <http://ccc.inaoep.mx/grodig/articulos.htm>
- Boyle, T. y Cook, J. 2001. "Towards a Pedagogically Sound Basis for Learning Object Portability and Re-Use". En The 18 th Annual Conference of the Australasian Society for Computers in Learning in Tertiary Education. Melbourne: Australia, December 9 to 12, 2001. <http://www.ascilite.org.au/conferences/melbourne01/pdf/papers/boylet.pdf>
- Del Moral, M. E. & Cernea, D. A., 2005, "[Diseñando Objetos de Aprendizaje como facilitadores de la construcción del conocimiento](#)". Disponible en <http://www.uoc.edu/symposia/spdece05/contribuciones.html>
- García Aretio, L., 2005, "[Objetos de Aprendizaje. Características y repositorios](#)". Disponible en <http://www.uned.es/catedraunesco-ead/editorial/p7-4-2005.pdf>
- Gibbons, A. S., Nelson, J. & Richards, R. 2000. "The nature and origin of instructional objects". In D. A. Wiley (Ed.), *The Instructional Use of Learning Objects*: Online Version. <http://reusability.org/read/chapters/gibbons.doc>

- Griffith, R. 2003. Learning Objects in Higher Education. Academic Advanced Distributed Learning CO-LAB (Academic ADL Co-Lab). http://www.academiccolab.org/resources/webct_learningobjects.pdf
- Gynn, C. y Acker, S.R. 2003. Learning Objects: Contexts and Connections. The Ohio State University. <http://www.elearningworkshops.com/docs/LO/LOResearch.pdf>
- Ip, A. y Morrison, I. 2001. "Learning Objects in Different Pedagogical Paradigms". En The 18 th Annual Conference of the Australasian Society for Computers in Learning in Tertiary Education. Melbourne: Australia, December 9 to 12, 2001. [http://users.tpg.com.au/adslfrcf/lo/LO\(ASCILITE2001\).pdf](http://users.tpg.com.au/adslfrcf/lo/LO(ASCILITE2001).pdf)
- Koper, R. 2003. "Combining re-Usable Learning Resources to Pedagogical Purposeful Units of Learning". In Allison Littlejohn and Simon Buckingham Shum (Eds.), Reusing Online Resources: A Sustainable Approach to eLearning. (Special Issue) Journal of Interactive Media in Education. London: Kogan Page. ISBN: 0749439491. <http://www-jime.open.ac.uk/2003/1/>
- Morales, E., García, F., Barrón, A., Berlanga, A. & López, C., 2005, "[Propuesta de Evaluación de Objetos de Aprendizaje](#)". Disponible en http://www.uoc.edu/symposia/spdece05/con_tribuciones.html
- Orrill, C. H. 2000. "Learning objects to support inquiry-based online learning". In D. A. Wiley (Ed.), The Instructional Use of Learning Objects: Online Version. <http://reusability.org/read/chapters/orrill.doc>

- Rebak, D. y Mason, R. 2003. "Keeping the Learning in Learning Objects". In Allison Littlejohn and Simon Buckingham Shum (Eds.), Reusing Online Resources: A Sustainable Approach to eLearning. (Special Issue) Journal of Interactive Media in Education. London: Kogan Page. ISBN: 0749439491. <http://www-jime.open.ac.uk/2003/1/>
- Sampedro Nuño, A., Sariego Ferrero, R., Martínez Nistal, A., Martínez González, R. A., Rodríguez Ruiz, B., 2005, "Procesos implicados en el desarrollo de materiales didácticos reutilizables para el fomento de la cultura científica y tecnológica", en [RED: Revista de Educación a Distancia](#), ISSN 1578-7680, N° 3. Disponible en <http://www.um.es/ead/red>
- Sánchez-Alonso, S. 2004. "Reutilización efectiva de los contenidos educativos mediante el diseño por contrato de objetos didácticos". Comunicación presentada en las V Jornadas de Informática y Sociedad (JIS 2004). ESIDE, Universidad de Deusto, Bilbao.
- Toews, J.; Reeser, S.; Johnson, A. 2002. How to Convert Content into a Sharable Content Object (SCO). Academic Advanced Distributed Learning CO-LAB (Academic ADL Co-Lab). <http://www.academiccolab.org/partners/docs/ConvertContentToSCO.pdf>
- Treviranus, J. y Brewer, J. 2003. "Developing and Reusing Accessible Content and Applications". In Allison Littlejohn and Simon Buckingham Shum (Eds.), Reusing Online Resources: A Sustainable Approach to eLearning. (Special Issue) Journal of Interactive Media in Education. London: Kogan Page. ISBN: 0749439491. <http://www-jime.open.ac.uk/2003/1/>

- Varas, M. L., 2002, "[Repositorios de Objetos de Aprendizaje](http://www.alejandria.cl/recursos/documentos/documentos_varas.doc)". Disponible en http://www.alejandria.cl/recursos/documentos/documentos_varas.doc
- Wiley, D. A. 2000. "Connecting learning objects to instructional design theory: A definition, a metaphor, and a taxonomy". In D. A. Wiley (Ed.), The Instructional Use of Learning Objects: Online Version. <http://reusability.org/read/chapters/wiley.doc>
- Williams, D. D. 2000. "Evaluation of learning objects and instruction using learning objects". In D. A. Wiley (Ed.), The Instructional Use of Learning Objects: Online Version. <http://reusability.org/read/chapters/williams.doc>

ANEXOS

Manual del desarrollador e instalación del Repositorio de Metadata

Con el objetivo de dar una introducción técnica a los desarrolladores que realizarán labores de mantenimiento y actualización del repositorio de metadata, a continuación se presenta una guía paso a paso con los elementos que conforman el sistema y como su incorporación.

Primer paso: Instalación de Axis2

Prerrequisitos

- Se debe tener instalado el Java Development Kit 1.5 (Se utilizó la versión 1.5 update 14 en el desarrollo de este repositorio).
- 11 Mb de espacio libre en disco.
- Sistema Operativo Windows XP, Linux, Mac OS X, Fedora core, Ubuntu, Gentoo

Instalación

Para instalar Axis2 se debe descargar del sitio web de Apache Project el archivo binario compatible con el ambiente Windows.

Para esto, se deben realizar los siguientes pasos:

1. Se debe ingresar al WebSite de Axis2/Java en Apache Project : <http://ws.apache.org/axis2/download.cgi>
2. Descargar el paquete en la versión 1.3. Seleccionando el Standard Binary Distribution en formato ZIP, ya que estamos trabajando en Microsoft Windows.
3. Descomprimir el archivo .ZIP en la raíz del disco duro principal (C:\). Esto debe crear una carpeta llamada C:\axis2-1.3
4. Configurar las siguientes variables de ambiente: `AXIS2_HOME = C:\axis2-1.3`
`JAVA_HOME = C:\Archivos de programa\Java\jdk1.5.14` (esta ruta puede cambiar según se haya realizado la instalación del JDK y/o la versión)
5. Se inicia un servidor standalone de Axis2, ejecutando el siguiente comando:

C:\axis2-1.3\bin\axis2server.bat

Después de esto, el servidor Web que trae por defecto Axis2, queda disponible a través del puerto http 8080.

Instalación de db4o

Prerrequisitos

- Descargar el motor db4o for Java de <http://www.db4o.com/community/>.
- Se debe tener instalado JRE v1.5 o superior.

Instalación

Para instalar db4o se deben descargar ejecutable compilado en binario, el cual es compatible con el ambiente Windows, plataforma en la cual se estará desarrollando y ejecutando el repositorio de metadata.

Para esto, se deben realizar los siguientes pasos:

1. Se debe ingresar al WebSite de db4o: <http://www.db4o.com/community/>.
2. Descargar el paquete en la versión 6.1. Seleccionando db4o 6.1 for Java.
3. Descomprimir el archivo .ZIP en la raíz del disco duro principal (C:\). Esto debe crear una carpeta llamada C:\db4o-6.1
4. Configurar la siguiente variable de ambiente en el sistema operativo:
CLASSPATH = C:\db4o-6.1\lib\db4o-6.1-java5.jar

Instalación y configuración de NetBeans con Axis2 y Apache Tomcat

Prerrequisitos

- NetBeans IDE versión 6.1.
- Java JDK versión 1.5 o superior.
- Apache Axis2.
- Apache Ant versión 1.7.
- Tomcat Web Server versión 6.0 o superior.
- Tener conexión a Internet activada.

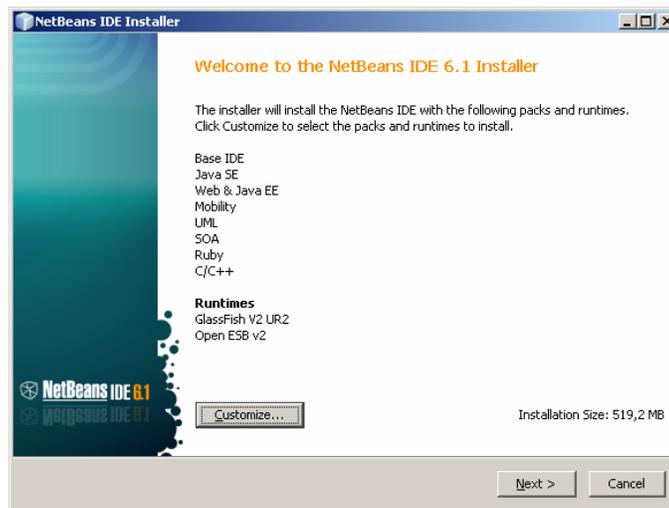
Instalación

Para instalar NetBeans IDE, primero se debe descargar de <http://download.netbeans.org/netbeans/6.1/final/> la versión full para Microsoft Windows (netbeans-6.1-windows.exe). Luego ejecutar el instalador y seguir los siguientes pasos (Los pasos descritos son las configuraciones necesarias para utilizar Netbeans IDE con Axis2 y Tomcat.):



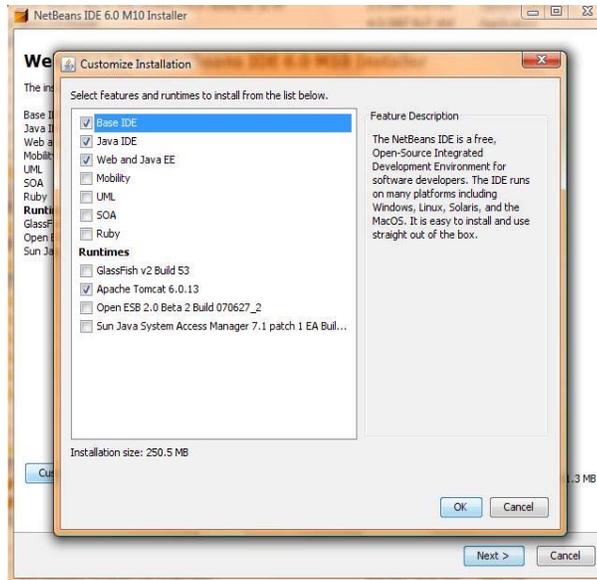
Inicio de instalación NetBeans IDE.

Luego, se muestra los componentes seleccionados por defecto para la instalación de NetBeans IDE. Se debe seleccionar el botón "Customize", para seleccionar las características que se necesitan.



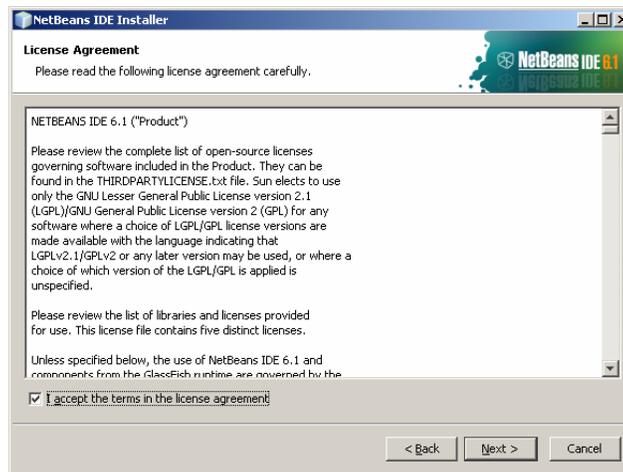
Componentes a instalar.

Se muestra los paquetes y ambientes disponibles que trabajan con esta versión de NetBeans IDE. Los paquetes que se deben seleccionar son: Base IDE, Java SE, Web & Java EE. Se debe seleccionar también el runtime de Apache Tomcat.



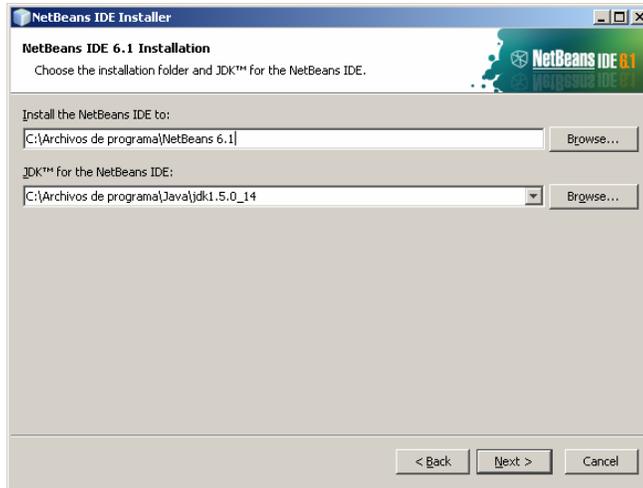
Selección de paquetes y ambientes.

Se muestra el acuerdo de licencia del producto (uso público), se debe aceptar los términos de los acuerdos para continuar con la instalación.



Acuerdo de Licencia.

Se muestra la ruta seleccionada para instalar NetBeans IDE y la ruta de instalación de Java.



Rutas de instalación NetBeans y Java.

Se muestra una alerta de seguridad en el caso de que este activado el Firewall de Windows XP SP2. Se debe seleccionar la opción “Desbloquear” para continuar con la instalación.



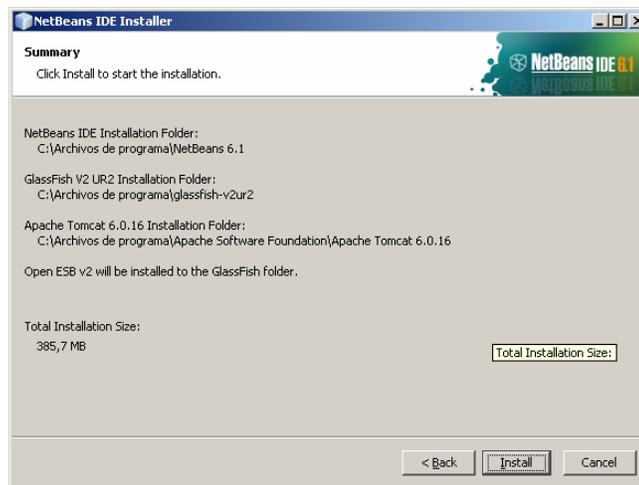
Alerta de seguridad de Windows.

Se muestra las opciones por defecto para la configuración de Apache Tomcat. En este caso, solo se pide la ruta de instalación.



Opciones de Apache Tomcat.

Se muestra un resumen con toda la configuración seleccionada para la instalación de NetBeans IDE. Al presionar el botón "Install", se comenzará a instalar automáticamente el producto.



Resumen de instalación NetBeans IDE.

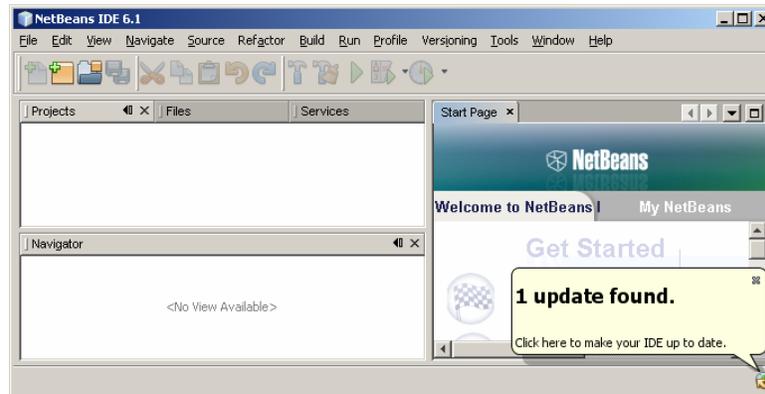
Configuración de Axis2 con NetBeans IDE

Luego de haber instalado el producto, se debe proceder a instalar y activar el plugin de Axis2 para desarrollar los Web Services. A continuación se presentan los pasos necesarios para su configuración.

Como primer paso se debe abrir NetBeans IDE; la primera vez que se abra el producto, este buscará en Internet las actualizaciones disponibles (recuerde que como

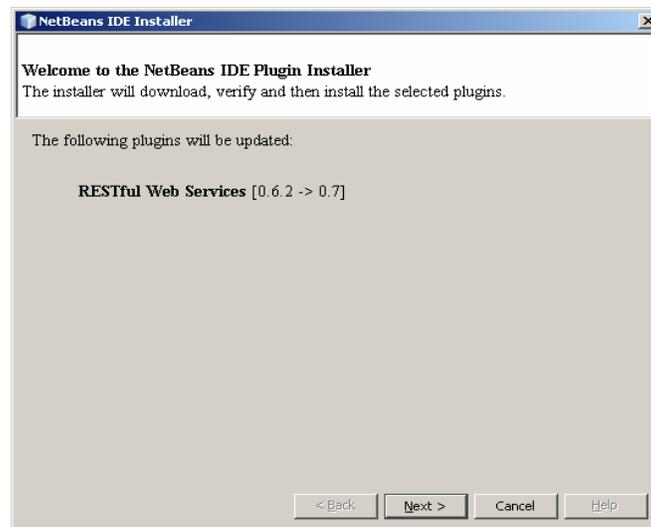
requisito se debe tener una conexión a Internet), en el caso de que encuentre una actualización disponible, NetBeans IDE lo informará.

Para instalar la actualización, sólo se debe presionar un click izquierdo del ratón sobre el mensaje y aparecerá una nueva ventana como muestra:

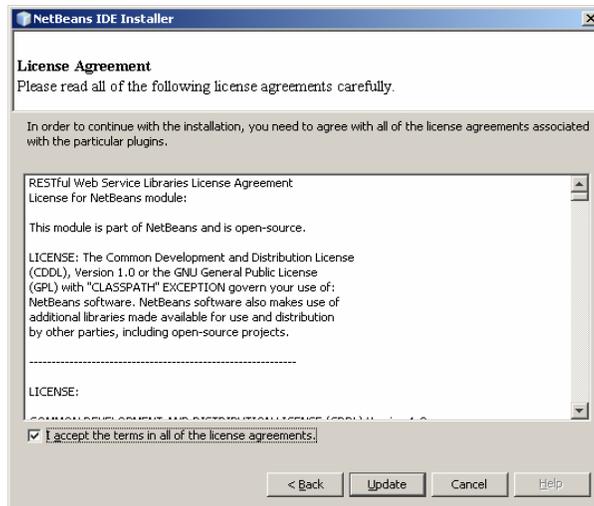


Actualizaciones disponibles para NetBeans IDE.

Se deberá esperar que NetBeans IDE termine de descargar desde Internet, la información necesaria para la instalación de la actualización; presionar siguiente. Luego, en la siguiente ventana aparecerá el acuerdo de licencia que se deberá aceptar para continuar con la instalación y presionar "Update", como muestra:

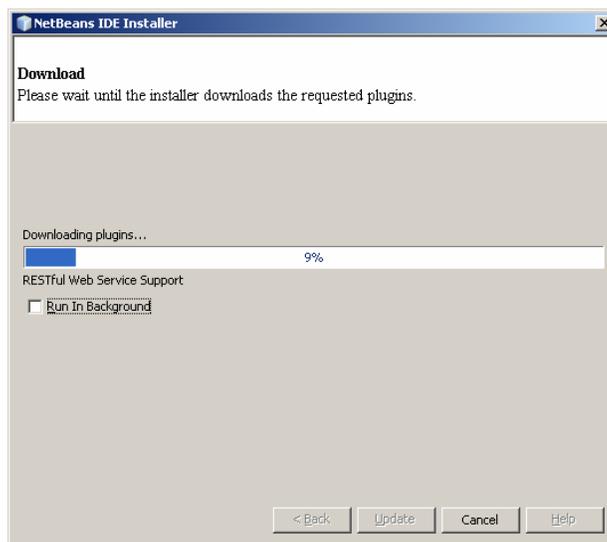


Instalación de actualizaciones disponibles para NetBeans IDE.



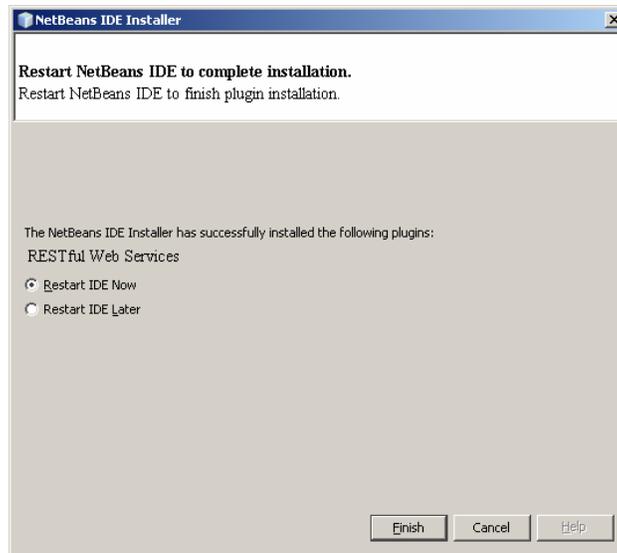
Acuerdo de Licencia para actualizaciones de NetBeans IDE.

Al presionar "Update", automáticamente se descargará e instalará la actualización, como se muestra:



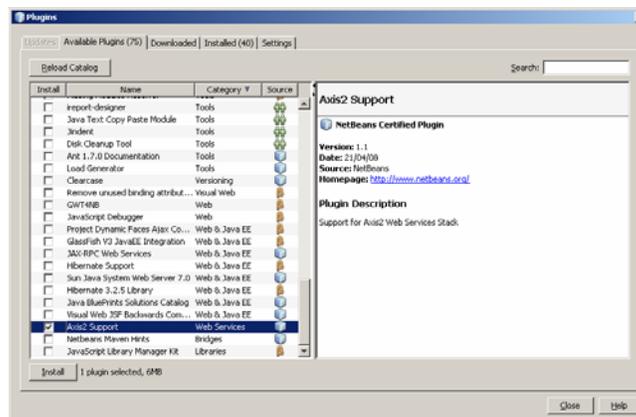
Progreso de la descarga e instalación para actualizaciones de NetBeans IDE.

Como paso final de esta parte, el instalador solicitará que reinicie el NetBeans IDE para aplicar las actualizaciones instaladas, presionar "Finish", como se muestra:



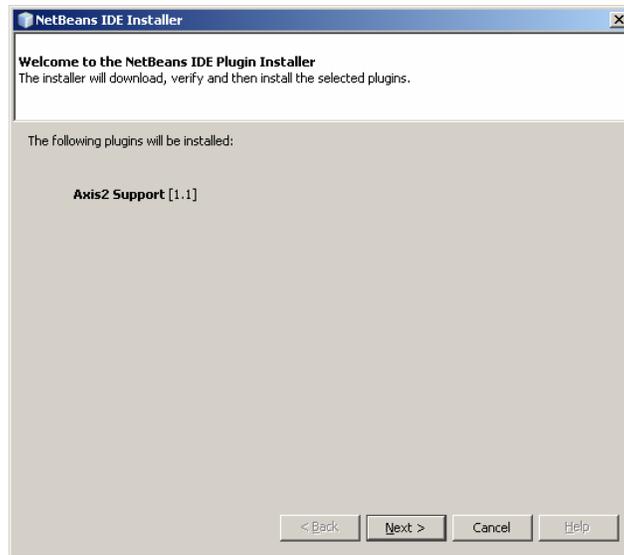
Finalizar instalación de actualizaciones de NetBeans IDE.

Después de que reinicie NetBeans IDE, seleccionar en la barra de menú “Tools” y luego “Plugins”. Allí mostrará una ventana de plugins disponibles, en la que se debe seleccionar “Axis2 Support” y presionar el botón “Install”:



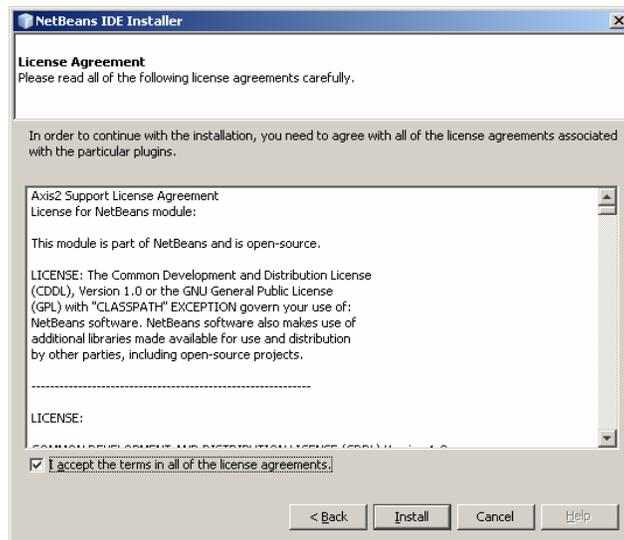
Instalación de Plugins para NetBeans IDE.

Seguidamente, aparecerá una nueva ventana que indica el inicio de la instalación del plugin “Axis2 Support” para NetBeans IDE:



Instalación de Axis2 Plugin para NetBeans IDE.

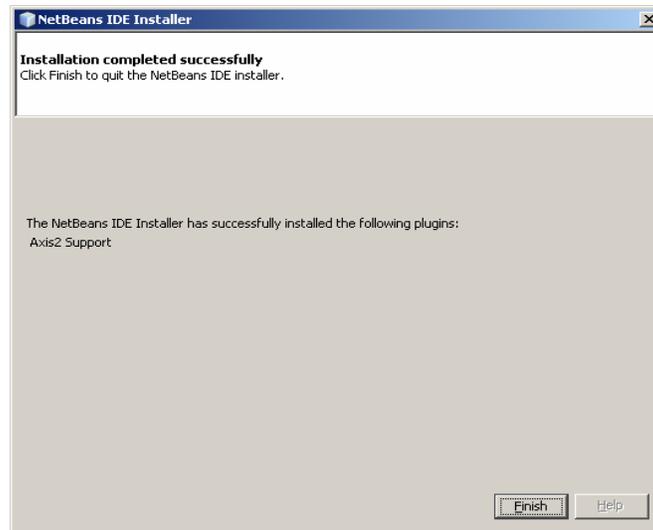
La próxima ventana muestra el acuerdo de licencia para la instalación de este plugin, se debe aceptar el acuerdo y presionar "Install", como se muestra en la siguiente figura:



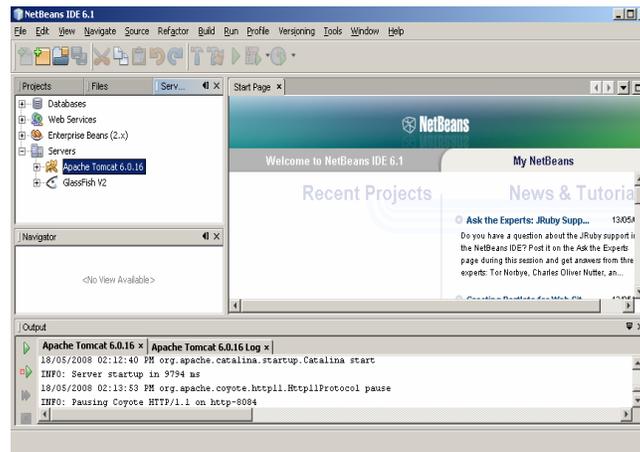
Acuerdo de Licencia para Axis2 Plugin.

La próxima ventana indica el progreso de la descarga e instalación de dicho plugin y luego mostrará una ventana de confirmación de la instalación. Como paso final de esta configuración, se debe reiniciar NetBeans IDE para que aplique los cambios. También se debe subir el servidor Tomcat, para que se cree el ambiente del servidor Web. En la pestaña de servicios, hay un renglón para los servidores en los que se

encuentra Apache Tomcat 6.0.16 con el botón derecho del ratón se presiona sobre el servidor y aparecen las opciones en la que se encuentra “Start” para iniciar el servicio.



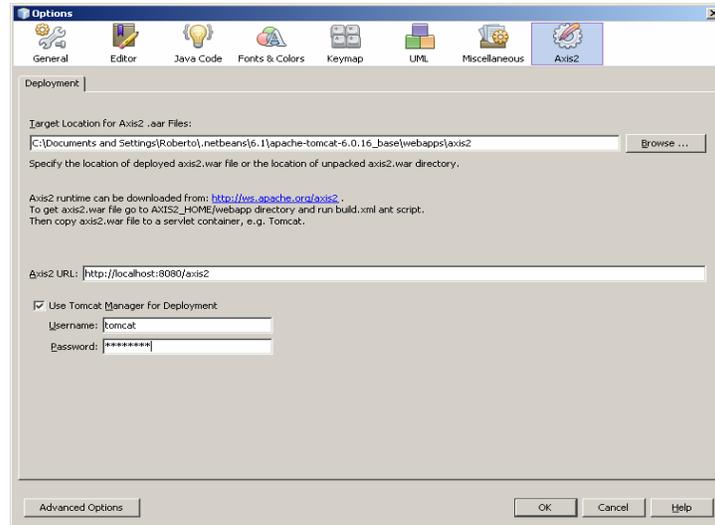
Fin de instalación de Axis2 Plugin.



Servicio Apache Tomcat.

Como paso siguiente, se debe copiar el archivo axis2.war, que está disponible para su descarga en el mismo sitio donde se descargan los archivos binarios de Axis2. Se debe copiar en la ruta por defecto donde se encuentra Tomcat para NetBeans, en nuestro caso la ruta por defecto es: C:\Documents and Settings\Usuario\.netbeans\6.1\apache-tomcat-6.0.16_base\webapps. Hecho esto, se procederá a configurar las opciones de Axis2 para Tomcat. Para comenzar, se debe seleccionar en la barra de menú “Tools” y luego “Options” y aparecerá una ventana de

las opciones de herramientas de NetBeans IDE, en la cual seleccionaremos el último icono en la parte superior derecha de dicha ventana. Allí aparecerán las opciones de Axis2, como se muestra:



Opciones de Axis2 Plugin.

Las opciones que se deben colocar en esta ventana son:

- **Target Location for Axis2 .aar Files.** En esta opción se indica donde se encontraran almacenados físicamente los servicios a desarrollar. La ruta que indicaremos en nuestro caso será: C:\Documents and Settings\Usuario\.netbeans\6.1\apache-tomcat-6.0.16_base\webapps\axis2.
- **Axis2 URL.** Esta opción indica la dirección URL donde podremos encontrar Axis2, es la siguiente: <http://localhost:8080/axis2>.
- **Use Tomcat Manager for Deployment.** Esta opción crea el enlace de NetBeans IDE con Apache Tomcat, para que el desarrollo de los Web Services sean configurados directamente en el Axis2 que corre bajo Tomcat.

Hecho esto, se debe reiniciar el servicio de Tomcat y quedará totalmente configurado Axis2 para trabajar con NetBeans IDE.

Manual del desarrollador e instalación de la Capa de Servicios de AMBAR

Herramientas utilizadas para el desarrollo de la Capa de Servicios de base de datos de AMBAR:

Kit de desarrollo Java (J2ee SDK)

- Descargar e instalar el jsdk 1.4.2 de la página web de sun (<http://developers.sun.com/downloads/>).
- Agregar la variable de entorno JAVA_HOME al sistema, cuyo valor será el directorio donde se ha instalado el jdk.

Kits de desarrollo de Servlets y JSP

- Descargar el software que implementa las especificaciones Java Servlet 2.1 o 2.2 y Java Server Pages 1.0 o 1.1 de la página web de sun (<http://java.sun.com/products/servlet/>).
- Indicar al javac donde encontrar las clases Servlets y JSP cuando se compile el archivo.

Servidor Apache Tomcat

- Descargar e instalar el binario de Tomcat en formato .exe de la página del grupo Jakarta (<http://jakarta.apache.org/>).
- Agregar una nueva variable de entorno TOMCAT_HOME al sistema, cuyo valor será la ruta del directorio donde se ha instalado el Tomcat.

Axis

- Descargar la versión binaria de axis de la página web (<http://ws.apache.org/axis/>), descomprimir el archivo y copiar el directorio de axis en el directorio de Tomcat/webapps/.

- Configurar las siguientes variables de entorno:
 - `AXIS_HOME`, con la ruta donde se ha copiado el axis.
 - `AXIS_LIB`, con el valor `%AXIS_HOME%\lib`.
 - `AXISCLASSPATH`, con el valor de todos los jar necesarios para su ejecución.

APIs adicionales

- Descargar los APIs, colocarlos en el directorio lib de la aplicación y añadirlos al "CLASSPATH":
 - Commons-fileupload-1.2
 - Commons-io-1.3.1
 - JAX-RPC