UNIVERSIDAD DE LOS ANDES
FACULTAD DE INGENIERÍA
DIVISIÓN DE ESTUDIOS DE POSTGRADO
DEPARTAMENTO DE SISTEMAS COMPUTACIONALES

Sistema Buscador de Contenidos Digitales de la Nube de Conocimiento del Proyecto Madre

Autor: Ricardo Dos Santos.
Tutor: PhD. José Aguilar.

Trabajo de grado presentado ante la Ilustre Universidad de Los Andes como requisito parcial para optar al grado de Magister Scientiae en Computación

Mayo 2015

©2014 Universidad de Los Andes Mérida, Venezuela

Resumen

En este trabajo se propone un sistema buscador de contenidos digitales adaptados a los estilos de aprendizajes, utilizando tecnologías como los servicios web para lograr mayor interoperabilidad entre sistemas, y ontologías para permitir razonar sobre el conocimiento. Este sistema brinda contenidos que coincidan con las herramientas, actividades e instrumentos, según el perfil educativo de los estudiantes. Para ello, consulta a un servicio web que provee información de los paradigmas de aprendizaje, y con esta información optimiza la búsqueda de los contenidos, logrando responder a la necesidad particular de un estudiante según su estilo de aprendizaje. Además se muestra la integración de este servicio web con el Proyecto Madre, donde interactúan un conjunto de servicios que garantizan un proceso de auto-formación, que va a ser ejecutado por el estudiante guiado por los profesores.

Palabras Clave

Contenidos digitales, Paradigmas de aprendizaje, Arquitectura guiada por Ontologías (ODA), Arquitectura orientada a servicios (SOA), Plataforma Web Educacional.

Índice general

I. Introducción **1**

1.1. Generalidades 1

1.2. Antecedentes 2

1.3. Objetivos 5

1.3.1. Objetivo General 5

1.3.2. Objetivos Específicos 5

1.4. Organización del trabajo 5

II. Marco Teórico **7**

2.1. Fundamentos Teóricos del Proyecto Madre 7

2.1.1. Definición del Proyecto Madre 7

2.1.2. Organización en Nubes 7

2.1.2.1. Nube de Auto-Formación 8

2.1.2.2. Nube de Paradigmas de Aprendizaje 9

2.1.2.3. Nube de Fuentes de Conocimiento 11

2.2. Servicios Web 12

2.2.1. Aplicaciones Orientadas a Servicios (SOA) 12

2.2.2. Definición de Servicios Web 14

2.2.2.1. Componentes de las Plataformas para Despliegue de Aplicaciones Orientadas a Servicios Web 14

2.2.2.2. Lenguaje descripción de servicios Web (WSDL) 16

2.2.2.3. Simple Object Access Protocol (SOAP) 17

2.2.2.4. Universal Description, Discovery and Integration (UDDI) . . 19

2.3. Ontologías 21

2.3.1. Definición de Ontologías 21
2.3.2. Componentes de las Ontologías 22
2.3.3. Clasificación de las Ontologías 22
2.3.4. Arquitectura Orientada por Ontologías (ODA) 23
2.3.4.1. Ingeniería dirigida por Modelos 23
2.3.4.2. Ingeniería dirigida por Ontologías 25
2.3.4.3. Metodología para el Desarrollo de aplicaciones basado en el paradigma ODA 26
2.4. Contenidos Digitales 28
2.4.1. Definición de Contenidos Digitales 28
2.4.2. Clasificación de los Contenidos Digitales 29
2.4.3. Ontology for media resources 1.0 de la W3C 30
2.4.4. Massive Open Online Course (Mooc) 31

III. Diseño del Sistema de Gestión de Contenidos Digitales 33
3.1. Arquitectura de Base del Proyecto Madre 33
3.2. Fase de Diseño 36
3.2.1. Conceptualización o Modelo Independiente de la Computación (CIM) 36
3.2.1.1. Requisitos funcionales 36
3.2.1.2. Requisitos no funcionales 42
3.2.1.3. Ontología Conceptual 42
3.2.2. Especificación o Modelo Independiente de la Plataforma (PIM) 46
3.2.2.1. Arquitectura Funcional 46
3.2.2.2. Ontología de Diseño 49

IV. Implementación del Sistema de Gestión de Contenidos Digitales 53
4.1. Arquitectura general del sistema 53
4.2. Fase de Desarrollo 54
4.2.1. Realización o Modelo Específico de la Plataforma (PSM) 54
4.2.1.1. Diagrama de Componente 54
4.2.1.2. Diagrama de Paquetes 57
4.2.1.3. Ontología de Desarrollo 59

V. Funcionamiento y Pruebas del Sistema **61**
5.1. Funcionamiento del sistema 61
5.1.1. El Proyecto Madre 61
5.1.2. Interfaz Para el Especialista 63
5.1.2.1. Interfaz Automatizada 63
5.1.2.2. Interfaz Genérica o Manual 64
5.1.2.3. Interfaz de Consultas 65
5.2. Pruebas de Sistema 66
5.2.1. Pruebas Unitarias 66
5.2.2. Pruebas de Integración 68
5.2.2.1. Interfaces Internas 69
5.2.2.2. Interfaces Externas 74
5.3. Comparación con otros trabajos 76

VI.Conclusiones y Recomendaciones **81**
6.1. Conclusiones 81
6.2. Recomendaciones 82

A. Caso de usos **85**
A.1. Registrar 85
A.2. Eliminar 85

Referencias Bibliográficas **87**

Índice de figuras

2.1. Paradigma de las Nubes. 8
2.2. Grafo Curricular. 9
2.3. Caracterización de las Dimensiones del Modelo de Felder-Silverman propuesta en [1] 11
2.4. Buscador de Objetos de Aprendizajes propuesto en [2] 12
2.5. Componentes de los Servicios Web 15
2.6. WSDL HelloWorld 17
2.7. Petición HTML 17
2.8. Respuesta HTML 18
2.9. Petición SOAP 18
2.10. Respuesta SOAP 19
2.11. Conversión WSDL a UDDI 20
2.12. UDDI HelloWorld 21
2.13. Ciclo de Vida del MDA 24
2.14. Fases de la Metodología 27
2.15. Ontology for media resources 1.0 30

3.1. Arquitectura Funcional de la Plataforma Web[3] 34
3.2. Proceso de invocación entre las nubes 35
3.3. Diagrama de Casos de Uso general para Los Contenidos Digitales 38
3.4. Métodos Consultar Interno y Externo 39
3.5. Diagrama de Actividades del proceso de Consultar Interno 40
3.6. Diagrama de Actividades del proceso de Consultar Externo 41
3.7. Ontología de Conceptualización de la Capa CIM de la plataforma 44
3.8. Ontology for media resources 1.0 46

3.9. Diagrama de Clases de la plataforma 48
3.10. Diagrama de Clases para la operación "ConsultarInterno" del servicio web . . 49
3.11. Ontología de Diseño de la Capa PIM de la plataforma 51

4.1. Arquitectura de la Plataforma Web 54
4.2. Diagrama de Componentes de la plataforma 56
4.3. Diagrama de Paquetes de la plataforma (Servidor) 58
4.4. Diagrama de Paquetes de la plataforma (Cliente) 58
4.5. Ontología de Desarrollo de la Capa PSM de la plataforma 60

5.1. Interfaz de la Nube de Auto-Formación 62
5.2. Ejemplo de Contenidos Recuperados por el Servicio de la Nube de Fuentes de Conocimiento desarrollado en este trabajo. 62
5.3. Interfaz Automatizada 64
5.4. Interfaz Manual 65
5.5. Interfaz de Consultas 66
5.6. Parte del código para la prueba automatizada con JUnit 67
5.7. Resultado de las pruebas automatizadas con JUnit 68
5.8. Método nubeAprendizaje del servicio web del sistema 69
5.9. Prueba de Integración con la Nube de Aprendizaje 70
5.10. Invocación del servicio web FContenidoDigital 70
5.11. Resultado de la prueba al servicio FContenidoDigital 71
5.12. Validaciones de los campos para las consultas 72
5.13. Resultados de las consultas 72
5.14. Interfaz de la Nube de Auto-Formación 75
5.15. Paradigma Activo 76

Índice de tablas

3.1. Caso de uso “Consultar” 37
3.2. Glosario de términos de la ontología CIM 43
3.3. Glosario de términos de la ontología de la Base de Datos 46
3.4. Glosario de términos de la ontología PIM 50

4.1. Glosario de términos de la ontología PSM 59

5.1. Contenidos digitales no certificados y sus características. 73
5.2. Contenidos digitales y sus características. 76
5.3. Tabla comparativa de sistemas desarrollados. 78

A.1. Caso de uso “Registrar” 85
A.2. Caso de uso “Eliminar” 86

Capítulo I

Introducción

1.1. Generalidades

En Venezuela se viene desarrollando un constante cambio a nivel educativo inspirada en los principios y en las conclusiones de la conferencia mundial sobre la educación superior realizada por la UNESCO (Organización de las Naciones Unidas para la Educación, la Ciencia y la Cultura) en octubre de 1998 en París. Se reconoce que hay, entre otras, una gran desigualdad en la distribución del conocimiento, y por lo tanto, con las posibilidades de desarrollo en los ciudadanos, por lo que se requiere de grandes cambios con respecto a la pertinencia del conocimiento impartido, determinado por su carácter social, la inclusión de sectores tradicionalmente excluidos de los sistemas educativos, y la flexibilización de las mallas curriculares para permitir mayor fluidez y libertad al proceso de formación académica.

Los Programas Nacionales de Formación (PNF), creado mediante Resolución 2.963 de fecha 13 de mayo de 2008, es el primer cambio rumbo a esa igualdad y flexibilidad, que queda plasmada en el currículo que forja los procesos de formación educativa en cuatro dimensiones principales: aprender a conocer, aprender a hacer, aprender a vivir (junto a aprender a convivir) y aprender a ser[4]. Los PNF se vienen dictando en los diferentes Institutos Universitarios de Tecnología (IUT) y los Colegios Universitarios (CU), que en el transcurso del tiempo el Gobierno Nacional, a través de la Misión Alma Mater, los ha transformado en las Universidades Politécnicas Territoriales (UPT).

Siguiendo estos cambio, la Universidad de Los Andes (ULA), a través del Proyecto Madre, busca formar parte de ésta transformación educativa, creando una carrera experimental denominada “Ingeniería en Computación e Informática”, empleando el paradigma de aprendizaje

basado en nubes. El paradigma lo conforman tres nubes que representan la auto-formación, los paradigma de aprendizajes, y las fuentes de conocimiento. Las nubes permiten una dinámica muy particular, donde se puede navegar de una nube a otra, ya que las fronteras no están claramente definidas y sus elementos pueden aparecer o desaparecer.

La singularidad de el paradigma basado en nubes requiere de cambios en todos los nivel de la institución, y uno de esos cambios radica en el sistema que acompañan a los profesores, estudiantes, y demás miembros de la comunidad, durante su estadía en la universidad. El sistema ya se encuentra en construcción, donde hay desarrollos en las diferentes nubes pero existe una necesidad latente en la nube de conocimiento, porque el actual desarrollo solo cubre objetos de aprendizajes, dejando por fuera contenidos como cursos en linea, guías, artículos y libros, entre otros.

Esta investigación pretende solucionar la necesidad de encontrar contenidos digitales generales que necesita la nube de conocimiento. Dichos contenidos están regados en la inmensa Internet, se hace necesario una búsqueda integrada al estilo de aprendizaje de los estudiantes, para ayudarlos en cada uno de los temas que conforman los módulos de la carrera.

1.2. Antecedentes

El primer trabajo titulado "Los sistemas de organización del conocimiento y el manejo de contenidos digitales"[5], trata sobre la organización y recuperación de contenidos digitales apoyándose en los sistemas de organización de conocimientos, los cuales actúan como puente entre la necesidad de información del usuario y las colecciones multimedia; y describe la clasificación de dichos sistemas. También destacan el uso de las ontologías en la administración de colecciones multimedia, en las que a cada archivo de imagen, vídeo o audio, les puede ser asociada información semántica que permita acceder a él de manera más rápida y precisa. Esa información semántica es, a grosso modo, la siguiente: autor, título, descripción, certificación (mecanismos de confianza y credibilidad que permitan determinar la calidad del contenido), entre otros atributos, que contribuyen a optimizar su localización.

Otro trabajo interesante es "Neptuno: tecnologías de la web semántica para una hemeroteca digital"[6], en el cual proponen la introducción de las tecnologías basadas en semántica para mejorar los procesos de creación, mantenimiento y explotación de la hemeroteca digital

de un diario. El aporte más significativo es la estructura de la plataforma, que consiste, primero en una base de conocimiento asociada a la hemeroteca, basada en una ontología para la descripción de la información; segundo, en un módulo de búsqueda semántica; y por ultimo, en un módulo de exploración y visualización de contenidos basado en ontologías. La ontología contiene, entre otros conceptos, los tomados de la clasificación temática del estandar IPTC (International Press Telecommunications Council). Además, el módulo de búsqueda semántica consiste en la combinación de las consultas directas por clases de contenidos y campos de las clases, con la posibilidad de explorar a través de la taxonomía IPTC la clasificación temática de las noticias y documentos de la hemeroteca.

En el trabajo "Aplicabilidad de ontologías en bibliotecas digitales"[7], conciben la Ontología de Informaciones Referenciales (OIR), con el objetivo de posibilitar un modelo conceptual que permite caracterizar los contenidos digitales, manteniendo referencias a los documentos académicos, contribuyendo a una mejor organización y recuperación de informaciones. Uno de los aportes es la construcción de la ontología OIR, basada en los estándares (MARC; Dublin Core y ETD-MS) y en el patrón Brasilero de Metadatos para Tesis y Disertaciones (MTD-BR). Esa ontología define, entre otros conceptos, diferentes clases que agrupan individuos de tipo libros, artículos, periódicos, trabajos, entre otros, permitiendo búsquedas con sentencias sencillas descritas en Lógica de Primer Orden, a partir de uniones se logran obtener reglas aun mas complejas.

Otro trabajado relacionado con el presente trabajo es "Búsqueda sobre catálogos basada en ontologías"[8], donde destaca la importancia de realizar búsquedas sobre catálogos a través de ontologías. Además, se exponen algunas de las metodologías, lenguajes y herramientas mayormente utilizadas en la elaboración de este tipo de representación de la información. El aporte más interesante para esta investigación es la ontología, que permite realizar búsquedas sobre un catálogo temático, es decir, sobre un conjunto de datos ordenados y clasificados, en este caso por temáticas. Esa ontología basa su criterio de búsqueda en función de conceptos y atributos; una vez establecida la consulta se navega por la estructura jerárquica que define la ontología para recuperar los términos que tengan relación con la petición realizada.

Por otro lado, en el trabajo "Uso das tecnologias na representação descritiva: o padrão de descrição bibliográfica semântica MarcOnt Initiative nos ambientes informacionais digitais"[9], se destaca la fuerte tendencia de los contenidos digitales y la falta de uso de estándares adecuados para la representación y descripción de estos recursos en los entornos de información digital. El aporte de esta investigación, es que detalla la iniciativa MarcOnt como un estándar para la representación y descripción de los contenidos, donde esta ontología almacenaría la información descrita en diferentes metadatos como BibTEX, BibTEXML, BibTEXRDF, Dublin Core, MARC21, MARCXML y MARCRDF. A grosso modo, proponen tomar al metadato MARC21 como base principal, y agregar los atributos de otros metadatos, buscando crear un estándar ontológico con capacidades de búsquedas semánticas.

Finalmente, las siguientes tres tesis implementan los diferentes módulos que conforman el paradigma de aprendizaje basado en nubes para el Proyecto Madre, cada una ellas despliega un servicio Web, que trabaja en conjunto con las demás, permitiendo un intercambio de información del estudiante según sus interés y su estilo de aprendizaje. Este trabajo desarrolla un nuevo modulo que se integra al proyecto.

En la primera tesis de grado, titulada "Prototipo de Plataforma Web para la Gestión de la Nube de Auto-formación Basado en Ontologías"[3], se desarrolla el servicio de la nube de formación la cual es la responsable de controlar e invocar los servicios que se desarrollarán en este trabajo, suministrando los parámetros con las necesidades de Contenidos Digitales para cada tipo de estudiante. La segunda, denominada "Implementar un prototipo de un sistema de gestión de la nube de aprendizaje usando el paradigma ODA"[1], toca todo lo relacionado a la nube de aprendizaje, la cual será el servicio a consultar para identificar el estilo de aprendizaje de un estudiante, parámetro que permite buscar contenidos adecuados para un estudiantes en particular. La ultima, "Plataforma basada en SOA usando el paradigma ODA (Ontology Driven Architecture), para la Gestión de una nube de fuentes de conocimiento compuesta por Objetos de Aprendizaje"[2], es la más relacionada con este trabajo, ya que en ella se desarrolla un Servicio Web para la nube de conocimiento que ofrece objetos de aprendizajes que permitan ser consultados por los estudiantes. Este trabajo deja un vacío en la nube de conocimiento, como son los contenidos generales como cursos, archivos PDF,

y cualquier otro tipo de contenido, que pueda ser parte de la información necesaria para el estudiante en su proceso formativo; dicho vacío es cubierto en esta investigación.

1.3. Objetivos

1.3.1. Objetivo General

- Desarrollar un sistema buscador de contenidos digitales, usando como insumo el estilo de aprendizaje de un estudiante, además del tema requerido a localizar.

1.3.2. Objetivos Específicos

- Caracterizar la información semántica que describe los Contenidos Digitales.
- Evaluar los estándares ontológicos que permitan el manejo de Contenidos Digitales.
- Identificar los métodos o funciones que gestionen un Contenido Digital a través del Servicio Web.
- Desarrollar e Integrar el servicio de Contenidos Digitales a la plataforma del proyecto madre.

1.4. Organización del trabajo

El documento se divide en seis partes. Este Capítulo 1, **Introducción**, ofrece una visión del contexto general de la investigación, que va desde las generalidades hasta los objetivos de la presente investigación. El Capítulo 2, **Marco Teórico**, define las bases teóricas de la propuesta, que consiste en el proyecto madre, las aplicaciones basadas en servicios Web, la ontología con su capacidad de modelar el conocimiento, y por último, los contenidos digitales y sus características; El Capítulo 3, **Diseño del Sistema de Gestión de Contenidos Digitales**, describe la construcción del sistema siguiendo la metodología para el desarrollo de aplicaciones basado en el paradigma de arquitectura orientada por ontología. El Capítulo 4, **Implementación del Sistema de Gestión de Contenidos Digitales**, detalla la implementación del sistema según la metodología, presentado el modelo especifico de la plataforma. El Capítulo 5, **Funcionamiento y Pruebas del Sistema**, muestra los casos de

experimentación para probar todas las funcionalidades del sistema. Finalmente, el Capítulo 6, **Conclusiones y Recomendaciones**, especifica los logros y las posibles mejoras en la investigación.

Capítulo II

Marco Teórico

2.1. Fundamentos Teóricos del Proyecto Madre

2.1.1. Definición del Proyecto Madre

El proyecto madre nace por la necesidad de compensar la carente calidad del proceso de aprendizaje de un estudiante, este proyecto se viene desarrollando en el Departamento de Computación de la Universidad de Los Andes. Las ideas planteadas se basan en la creación de una escuela experimental en el ámbito de las Tecnologías de la Información, utilizando el paradigma de las nubes. Basada en dicho paradigma, la misma tiene características fuera de lo convencional, como por ejemplo, una dinámica centrada en los grupos de investigación, mallas curriculares adaptables a varios perfiles de formación profesional en TICs, y un modelo pedagógico basado en el paradigma "aprender haciendo", todo esto garantizando un proceso de auto-formación basada en recursos educativos, que va a ser ejecutado por el estudiante guiado por los profesores[10].

2.1.2. Organización en Nubes

El paradigma de aprendizaje basado en nubes lo conforman tres nubes que representan la auto-formación, los paradigmas de aprendizaje y las fuentes de conocimiento, como se observa en la figura 2.1; las nubes permiten una dinámica muy particular, donde se pueden navegar de una nube a otra, ya que las fronteras no están claramente definidas y sus elementos pueden aparecer o desaparecer, así como las interrelaciones entre ellos[10].

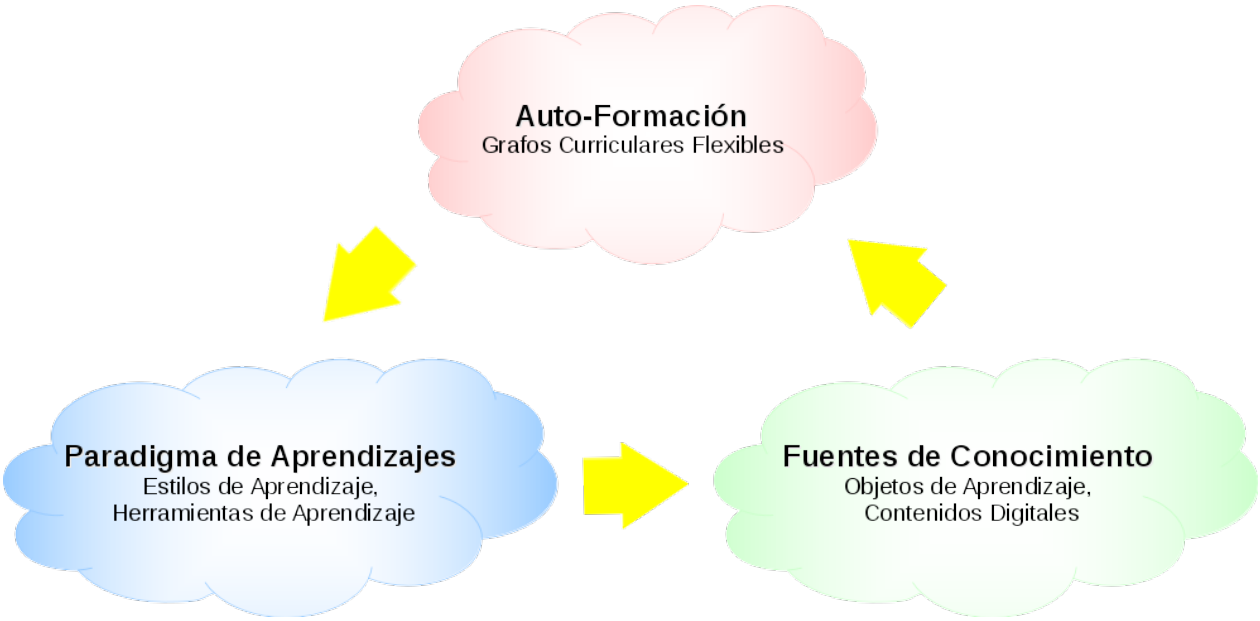

Figura 2.1: Paradigma de las Nubes.

2.1.2.1. Nube de Auto-Formación

La nube de auto-formación es el eje principal de la vida del estudiante, con toda la información proveniente de las nubes de los paradigmas de aprendizajes y de las fuentes de conocimiento. El paradigma de base es la inclusión de todos, entendiéndose por eso que cualquiera puede entrar a la nube de auto-formación. Quien entre a la nube de auto-formación debe estar consciente de los pre-requisitos exigidos por los módulos de las materias, que pueden ser cubiertos por cursos previos, pueden haber sido adquiridos por procesos de auto-formación, o eventualmente, pueden ser cubiertos durante la realización del módulo.

Esta nube permite al estudiante construir la malla curricular según su interés, logrando así conseguir competencias especificas y titulaciones en diferentes niveles: ingeniero, técnico medio, entre otros. Los perfiles de formación definidos para el recorrido del estudiante fueron inspirados en la curricula definida por la IEEE/ACM, como se muestra en la figura 2.2, donde se especifican 5 perfiles curriculares/profesionales, que son[11]: Ciencias de la Computación, Ingeniería del Software, Ingeniería del Computador, Sistemas de Información y Tecnologías de Información.

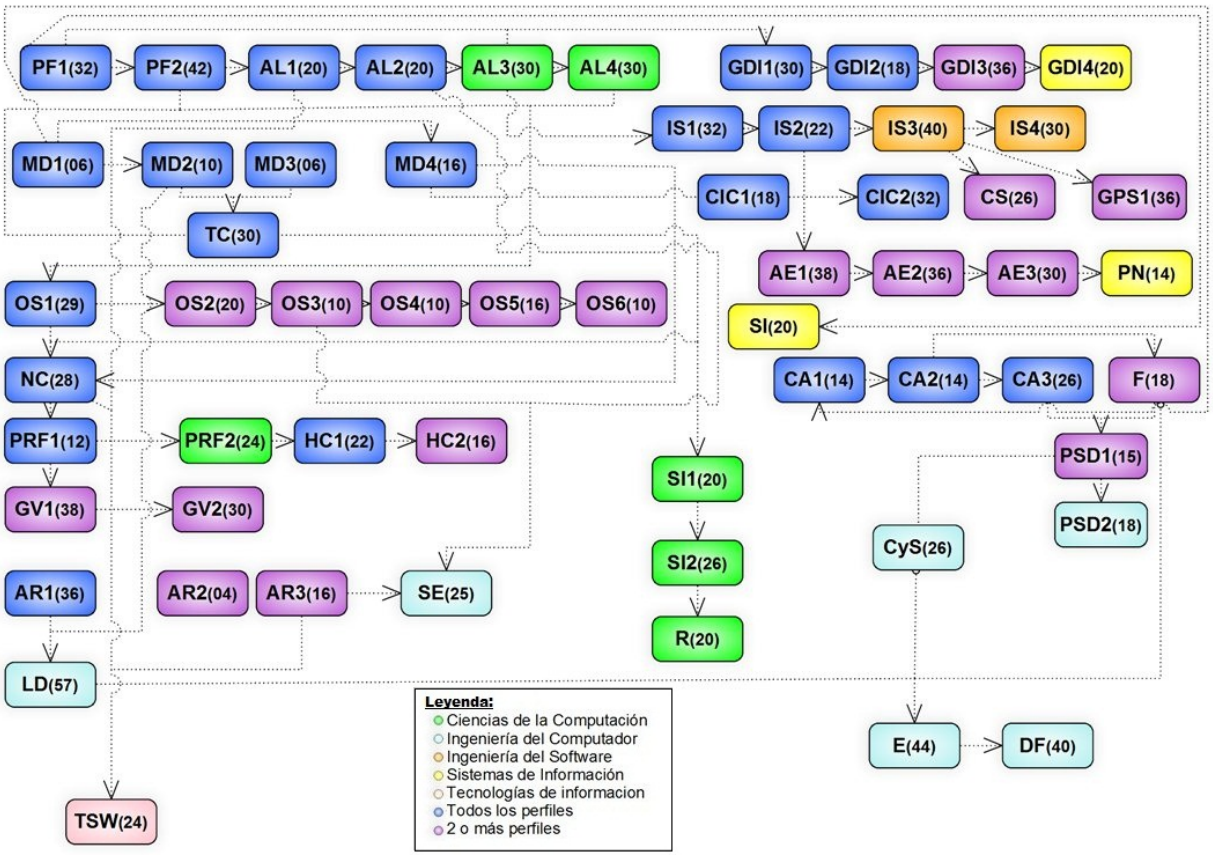

Figura 2.2: Grafo Curricular.

Por otro lado, hay que recalcar que el modelo de aprendizaje en la nube de auto-formación está basado en el desarrollo de obras durante el proceso de aprendizaje, en las cuales el estudiante va plasmando el conocimiento adquirido.

2.1.2.2. Nube de Paradigmas de Aprendizaje

En la nube de los paradigmas de aprendizaje es donde recae la responsabilidad de aportar los mecanismos necesarios para la auto-formación, según el estilo de aprendizaje del estudiante. Dicho estilo de aprendizaje determinará las estrategias y herramientas que mejor se adapten a la capacidad y aptitud de un estudiante particular. Esta nube guiará las dinámicas de auto-formación, establecerá formas de acreditar cursos, posibilitará maneras de plasmar los conocimientos adquiridos, espacios de intercambio, de trabajo colaborativo, de construcción colectiva del conocimiento, entre otras cosas. Permitirá navegar entre el mundo presencial y

virtual en el proceso de aprendizaje, garantizando elementos humanísticos en el proceso de aprendizaje.

Un estudiante posee un estilo de aprendizaje representado por un número de cuatro dígitos, cada dígito de este número es una dimensión según el modelo pedagógico de Felder-Silverman. Las cuatros dimensiones en las que se clasifica a los perfiles de aprendizajes de los estudiantes son[1]:

Activos o Reflexivos

Activos: Tienden a retener y comprender mejor nueva información cuando hacen algo activo con ella (discutiéndola, aplicándola, explicándosela a otros). Prefieren aprender ensayando y trabajando con otros.

Reflexivos: Tienden a retener y comprender nueva información pensando y reflexionando sobre ella, prefieren aprender meditando, pensando y trabajando solos.

Sensitivos o Intuitivos

Sensitivos: Concretos, prácticos, orientados hacia hechos y procedimientos; les gusta resolver problemas siguiendo procedimientos muy bien establecidos; tienden a ser pacientes; gustan de trabajo práctico (por ejemplo, trabajo de laboratorio); memorizan hechos con facilidad; no gustan de cursos a los que no les ven conexiones inmediatas con el mundo real.

Intuitivos: Conceptuales; innovadores; orientados hacia las teorías y los significados; les gusta innovar y odian la repetición; prefieren descubrir posibilidades y relaciones; pueden comprender rápidamente nuevos conceptos; trabajan bien con abstracciones y formulaciones matemáticas; no gustan de cursos que requieren mucha memorización o cálculos rutinarios.

Visuales o Verbales

Visuales: En la obtención de información prefieren representaciones visuales, diagramas de flujo, diagramas, etc.; recuerdan mejor lo que ven.

Verbales: Prefieren obtener la información en forma escrita o hablada; recuerdan mejor lo que leen o lo que oyen.

Secuenciales o Globales

Secuenciales: Aprenden en pequeños pasos incrementales, ordenados y lineales; tal que el siguiente paso está siempre lógicamente relacionado con el anterior; cuando tratan de solucionar un problema tienden a seguir caminos por pequeños pasos lógicos.

Globales: Aprenden en grandes saltos, aprendiendo nuevo material casi al azar y "visualizando la totalidad; pueden resolver problemas complejos rápidamente, y ponen juntas cosas en forma innovadora. Pueden tener dificultades, sin embargo, en explicar cómo lo hicieron.

En trabajos posteriores se han clasificado esas dimensiones en los siguientes rangos[1]: perfil alto (los valores 1 y 5), perfil mediano (los valores 2 y 4) y perfil equilibrado (el valor 3) como se muestra en color rojo en la figura 2.3. Estos perfiles permiten obtener el paradigma, los instrumentos, las herramientas y las actividades más acorde al estudiante.

	A						B						
	11	9	7	5	3	1	1	3	5	7	9	11	
ACTIVO	1		2		3				4		5		REFLEXIVO
SENSORIAL	1		2		3				4		5		INTUITIVO
VISUAL	1		2		3				4		5		VERBAL
SECUENCIAL	1		2		3				4		5		GLOBAL

Figura 2.3: Caracterización de las Dimensiones del Modelo de Felder-Silverman propuesta en [1]

2.1.2.3. Nube de Fuentes de Conocimiento

Por último, en la nube de fuentes de conocimiento confluyen las diferentes fuentes de información distribuidas a lo largo y ancho de la Internet. Ese conocimiento que se le entrega al estudiante esta adaptado a su perfil, primero basado en los requerimientos establecidos en la nube de auto-formación (por ejemplo, módulos a cursar), y segundo basado en los instrumentos, las actividades, y las herramientas especificadas en la nube de paradigma de aprendizajes para el estilo de aprendizaje del estudiante. Esta nube requerirá de herramientas para organizar, analizar, explotar, criticar, dicho conocimiento (es decir, de gestión del conocimiento), haciendo énfasis en la adquisición de conocimientos, en las destrezas para usarlo y reflexionar

sobre él, y en la actitud como ciudadano para reconocerlo en el contexto social donde este inmerso.

Actualmente esta nube posee conocimientos sobre objetos de aprendizajes (ver figura 2.4), gracias a la investigación desarrollada por Portilla[2], pero esta fuente de conocimiento se queda corta con la gran gama de información que esta dispersa en la Internet, debido a esto, la presente investigación amplia de manera general la capacidad de albergar cualquier contenido digital.

Figura 2.4: Buscador de Objetos de Aprendizajes propuesto en [2]

2.2. Servicios Web

2.2.1. Aplicaciones Orientadas a Servicios (SOA)

Últimamente, cuando se habla de Servicios Web se emplea también el término SOA, ¿Qué es? SOA es un acrónimo de Arquitectura Orientada al Servicio (Service Oriented Architecture, en inglés). SOA es un método para diseñar y construir software muy independiente (poco acopladas). Las funcionalidades son descritas como servicios, y son accesibles programáticamente por otras aplicaciones, a través de interfaces publicadas que puedan ser descubiertas.

2.2 Servicios Web

Los Servicios Web representan una implementación de una Arquitectura Orientada al Servicio.

En general, una arquitectura orientada al servicio es una colección de servicios. Estos servicios se comunican entre ellos. La comunicación puede involucrar simplemente el pase de datos o la coordinación de alguna actividad entre varios servicios.

Según Thomas Erl[12], los principios asociados a la Orientación a Servicios son:

Los Servicios deben ser reusables: Thomas menciona que todo servicio debe ser diseñado y construido pensando en su reutilización dentro de la misma aplicación, dentro del dominio de aplicaciones de la empresa, o incluso dentro del dominio público para su uso masivo.

Los Servicios deben proporcionar un contrato formal: Todo servicio desarrollado debe proporcionar un contrato en el cual figuren: el nombre del servicio, forma de acceso, las funcionales que ofrece, los datos de entrada de cada una de las funcionalidades y los datos de salida. De esta manera, todo consumidor del servicio accederá a este mediante el contrato, logrando así la independencia entre el consumidor y la implementación del propio servicio. En el caso de los Servicios Web, esto se logrará mediante la definición de interfaces con WSDL (Web Services Description Language).

Los Servicios deben tener bajo acoplamiento: Los servicios tienen que ser independientes los unos de los otros. Para lograr ese bajo acoplamiento, lo que se hará es que cada vez que se vaya a ejecutar un servicio, se accederá a él a través del contrato, logrando así la independencia entre el servicio que se va a ejecutar y el que lo llama.

Los Servicios deben permitir la composición: Todo servicio debe ser construido de tal manera que pueda ser utilizado para construir servicios genéricos de más alto nivel, el cual estará compuesto de servicios de más bajo nivel.

Los Servicios deben de ser autónomos: Todo servicio debe ser mantenido, desarrollado, instalado y versionado de forma independiente.

Los Servicios no deben tener estado: Un servicio no debe guardar ningún tipo de información, es decir, cada petición del cliente debe contener toda la información necesaria

para que el servicio la comprenda y no necesite mirar otro dato almacenado previamente sobre el contexto de la comunicación. La solución es que un servicio sólo contenga lógica, y que toda información esté almacenada en algún sistema de información, sea del tipo que sea.

Los Servicios deben poder ser descubiertos: Todo servicio debe poder ser descubierto de alguna forma para que pueda ser utilizado por otra aplicación, es decir, un consumidor de servicios verifica la localización dinámica de un servicio consultando el registro de servicios, para hallar uno que cumpla con un determinado criterio. Si el servicio existe, el registro proporciona al consumidor la interfaz de contrato y la dirección del servicio proveedor. En el caso de los Servicios Web, el descubrimiento se logrará publicando las interfaces de los servicios en registros UDDI (Universal Description, Discovery, and Integration).

2.2.2. Definición de Servicios Web

Los Servicios Web permiten la comunicación entre diferentes máquinas, con diferentes plataformas, entre programas distintos. Esta comunicación se logra a través de diversos estándares abiertos.

Según W3C[13]

> "Un servicio Web es un sistema de software diseñado para soportar la interacción interoperable máquina a máquina sobre una red. Cuenta con una interfaz descrita en un formato procesable por una máquina (específicamente WSDL). Además, los sistemas interactúan con el servicio Web de una manera prescrita por su descripción utilizando mensajes SOAP, transmitiendo por lo general por medio de HTTP con una serialización XML"

2.2.2.1. Componentes de las Plataformas para Despliegue de Aplicaciones Orientadas a Servicios Web

Los elementos básicos que conforman un servicio web son[14]: SOAP como lenguaje de intercambio, WSDL como lenguaje para la descripción de los servicios y UDDI para la publicación o registro de los mismos.

En el figura 2.5 se puede observar la existencia de tres roles claramente diferenciados:

Solicitante del servicio: Es el que solicita la ejecución del servicio web, y por lo tanto el que lo consume.

Proveedor del servicio: Es el encargado de implementar el servicio web y ofrecerlo a los clientes.

Directorio: Es un repositorio donde se almacenan las descripciones de los servicios, para que así los clientes puedan buscar el servicio web que mejor se adapte a sus necesidades.

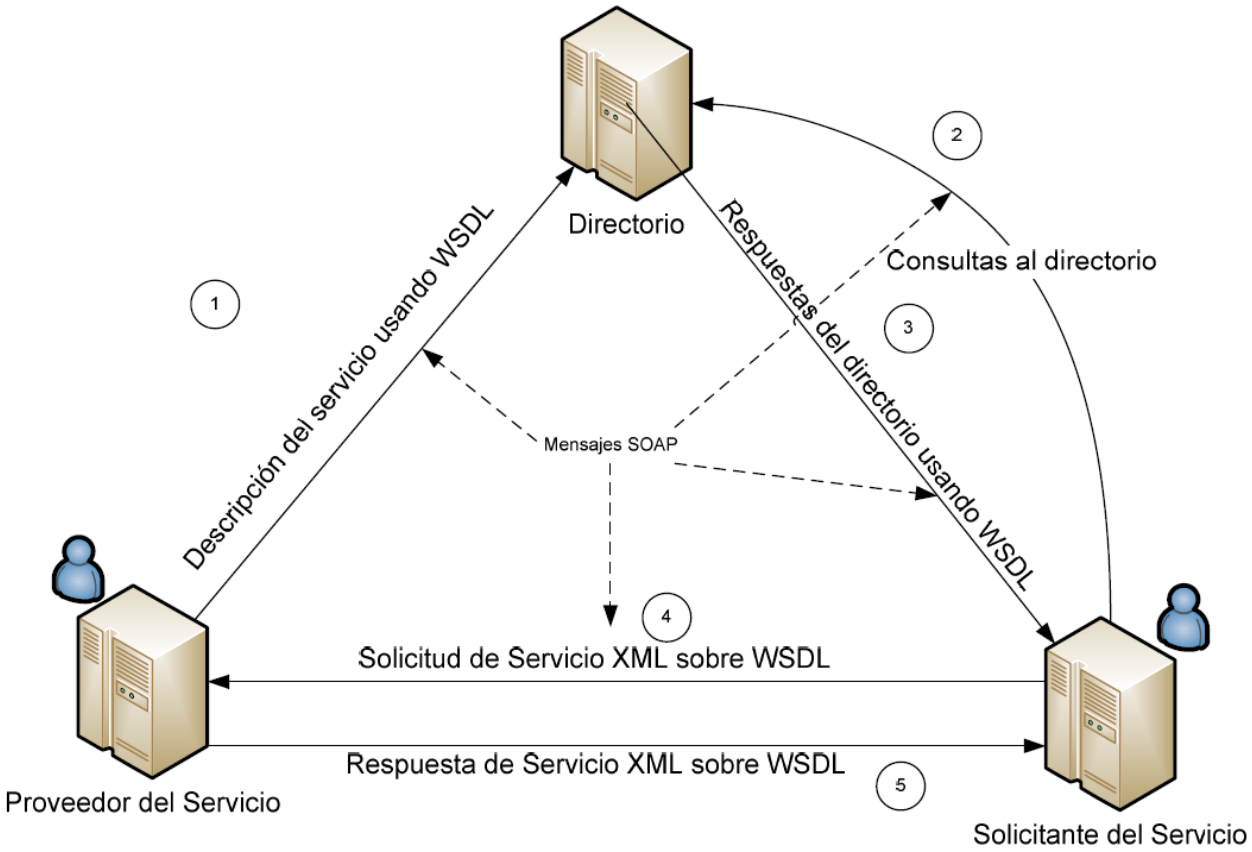

Figura 2.5: Componentes de los Servicios Web

La secuencia de ejecución entre los diferentes componentes es la siguiente (ver figura 2.5):

1. Un proveedor de servicios describe su servicio mediante WSDL. Esta definición se publica en un directorio de servicios utilizando UDDI.

2. Un solicitante del servicio emite una o más consultas al directorio para localizar un servicio y determinar la forma de comunicarse con ese servicio.

3. El directorio proporciona la descripción del servicio del proveedor al solicitante del servicio.

4. El solicitante del servicio utiliza la descripción del servicio para enviar una solicitud al proveedor de servicios.

5. El proveedor de servicios proporciona la respuesta esperada a un solicitante del servicio.

2.2.2.2. Lenguaje descripción de servicios Web (WSDL)

WSDL es un lenguaje basado en XML que se utiliza para describir un Servicio Web. Ha sido definido por la W3C como el estándar fundamental para la descripción de servicios. Un archivo con formato WSDL provee información de los distintos métodos (operaciones) que el Servicio Web brinda, muestra cómo accederlos y que formatos deben de tener los mensajes que se envían y se reciben. Es como un contrato entre el proveedor del servicio y el cliente, en el cual el proveedor se compromete a brindar ciertos servicios solo si el cliente envía un requerimiento con determinado formato. Es el documento principal a lo hora de documentar un Servicio Web, pero puede no ser el único. En la mayoría de los casos, es conveniente que este acompañado por un documento escrito en lenguaje natural, que brinde información de que es lo que hace cada uno de los métodos brindados por el Servicio Web.

Un archivo con formato WSDL básicamente contiene los siguientes elementos[15]:

Type: Describe los tipos no estándar usados por los mensajes (Message).

Message: Define los datos que contienen los mensajes pasados de un punto a otro.

Operation: Define una descripción abstracta de la acción soportada por el servicio.

PortType: Define una colección de operaciones brindadas por el servicio. Cada operación tiene un mensaje de entrada y uno de salida que se corresponde con algún Message antes definido.

Binding: Describe los protocolos que se utilizan para llevar a cabo la comunicación en un determinado PortType; actualmente los protocolos soportados son SOAP, HTTP GET, HTTP POST y MIME, siendo SOAP el más utilizado.

Port: Define una dirección (URL) para un determinado Binding

Service: Define una colección de Ports.

En el ejemplo WSDL HelloWorld de la figura 2.6, se muestra la implementación de los elementos Message y PortType, donde se indica los parámetros de entrada y salida del método u operación HelloWord.

```
<message name="HelloWorldSoapIn">
    <part name="parameters" element="s0:HelloWorld" />
</message>
<message name="HelloWorldSoapOut">
    <part name="parameters" element="s0:HelloWorldResponse" />
</message>
<portType name="Service1Soap">
      <operation name="HelloWorld">
            <input message="s0:HelloWorldSoapIn" />
            <output message="s0:HelloWorldSoapOut" />
      </operation>
</portType>
```

Figura 2.6: WSDL HelloWorld

2.2.2.3. Simple Object Access Protocol (SOAP)

SOAP (siglas de Simple Object Access Protocol) es un protocolo para el intercambio de información en un ambiente descentralizado y distribuido. Es el protocolo más utilizado para realizar el intercambio de información en los Servicios Web. Esta basado en XML, y potencialmente puede ser utilizado en combinación con una variedad de protocolos de comunicación, siendo el más utilizado HTTP. En los Servicios Web, se utiliza HTTP (figura 2.7) para transportar la información y XML para representar la misma.

```
POST /HolaInternetService/Service1.asmx/HelloWorld HTTP/1.1
Host: localhost
Content-Type: application/x-www-form-urlencoded
Content-Length: length
```

Figura 2.7: Petición HTML

En la figura 2.7 se observa en la primera línea la invocación al método HelloWorld a través de una petición SOAP utilizando el protocolo HTML, y retorna como respuesta la palabra "string" (figura 2.8).

```
HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<string xmlns="http://tempuri.org/">string</string>
```

Figura 2.8: Respuesta HTML

El modelo de comunicación de SOAP (figura 2.9) es muy similar al de HTTP. Un cliente hace un requerimiento (request), el servidor que esta escuchando los requerimientos lo atiende y responde (response), brindando la información solicitada o enviando un mensaje de error en caso de que el requerimiento no haya sido válido. El mensaje SOAP consiste en una envoltura obligatorio, un encabezado opcional, y un cuerpo obligatorio como un documento XML. El encabezado SOAP es utilizado para definir información acerca del requerimiento, mientras que el cuerpo SOAP contiene el método llamado y los parámetros con los que se llama al mismo.

```
POST /HolaInternetService/Service1.asmx HTTP/1.1
Host: localhost
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://tempuri.org/HelloWorld"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope          xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance"               xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <HelloWorld xmlns="http://tempuri.org/" />
  </soap:Body>
</soap:Envelope>
```

Figura 2.9: Petición SOAP

La petición (figura 2.9) y respuesta (figura 2.10) SOAP son más complejas que las HTML, ya que es un protocolo pensado para representar en modo XML tanto las llamadas a los métodos como sus respuestas.

```
HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope         xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance"              xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <HelloWorldResponse xmlns="http://tempuri.org/">
      <HelloWorldResult>string</HelloWorldResult>
    </HelloWorldResponse>
  </soap:Body>
</soap:Envelope>
```

Figura 2.10: Respuesta SOAP

2.2.2.4. Universal Description, Discovery and Integration (UDDI)

UDDI es un estándar para registrar y descubrir servicios web. La idea es que las distintas empresas registran su información acerca de los servicios web que provee, para que puedan ser descubiertas y utilizadas por potenciales usuarios. La información es ingresada al registro del directorio, un servicio lógicamente centralizado y físicamente distribuido a través de múltiples nodos, los cuales replican su información en forma regular. Una vez que un proveedor de servicio se registra en un determinado nodo del registro de directorio UDDI, la información es replicada a los otros nodos, y queda disponible para ser descubierta por cualquier solicitante de servicio.

El modelo de información base utilizado por los directorios es definido en un esquema XML. Este esquema define cuatro tipos básicos de información, cada uno de los cuales proveen la clase de información que un usuario necesita saber para utilizar un servicio web de un proveedor de servicio. Los cuatro tipos de información son[16]:

Información del negocio. Este tipo de información se encuentra definido dentro del elemento businnessEntity. Contiene información de la empresa, como su nombre, los contactos, el tipo de empresa, etc.

Información del servicio. Dentro del elemento businnessEntity se encuentran los elementos businessServices, estos elementos contienen información técnica e información sobre la descripción de los servicios web, generalmente agrupados por procesos de negocio o categorías de servicios.

Información del enlace. Dentro de cada elemento businessServices se encuentran los elementos bindingTemplate, las cuales son plantillas que enlazarán el servicio web con su implementación actual. Se puede enlazar cada servicio con un número cualquiera de plantillas (Binding Templates). Por lo tanto, una plantilla debe contener dos piezas importantes de información. Primero, una dirección física del lugar donde se encuentra la implementación actual, y en segundo lugar, la descripción acerca de su arquitectura.

Información sobre las especificaciones del servicio. Cada bindingTemplate tiene asociado un tModel, el cual brinda información sobre las especificaciones del servicio, por ejemplo, como tienen que ser los mensajes que el servicio espera y responde, etc. Un tModel puede ser asociado con elementos bindingTemplate a distintos proveedores de servicio que brindan la misma especificación del servicio. Utilizando los tModels se pueden encontrar todos los proveedores de servicio que brindan un servicio especifico.

Los directorios ofrecen búsqueda de servicios web a través de sus archivos UDDI para los usuarios o solicitante de servicio, gracias a la conversión del archivo WSDL del servicio web que registra un proveedor de servicio, en la figura 2.11 se puede observar esa conversión.

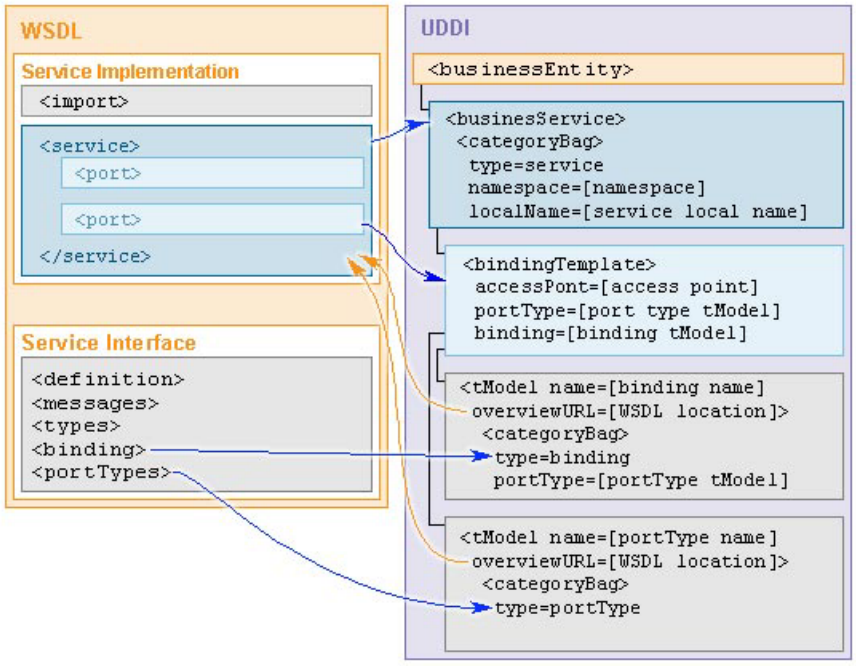

Figura 2.11: Conversión WSDL a UDDI

La figura 2.12 muestra parte del archivo UDDI HelloWorld, específicamente la sección tModel proveniente de portType del WSDL, donde el nombre tModel es el mismo que el nombre portType. tModel contiene la etiqueta categoryBag que especifica el espacio de nombres WSDL, e indica que tModel es de tipo "portType". Por otro lado, la etiqueta overviewDoc proporciona la dirección al documento WSDL.

```
<TModel TModelKey="uuid:e8cf1163-8234-4b35-865f-94a7322e40c3" >
    <name> HelloWorld Interface Port Type</name>
    <description>
        An interface for a friendly WebService
    </description>
    <overviewDoc>
        <overviewURL>
            http://localhost/helloworld.wsdl
        </overviewURL>
    </overviewDoc>
    </categoryBag>
        <keyedReference
            tModelKey="uuid:6e090afa-33e5-36eb-81b7-1ca18373f457"
            keyName="WSDL type"
            keyValue="portType" />
    </categoryBag>
</TModel>
```

Figura 2.12: UDDI HelloWorld

2.3. Ontologías

2.3.1. Definición de Ontologías

En la Actualidad existen diversas definiciones del concepto de ontología en el ámbito tecnológico, una de las más ampliamente aceptadas es la propuesta por Gruber (1993): "Una ontología es la especificación de una conceptualización"[17]. Luego, Gruber (2008) la detalla más explicando que “La ontología define un conjunto de primitivas de representación con la que se puede modelar un dominio de conocimiento”[18], sugiriendo que la ontología permite modelar el conocimiento, apoyándose en conceptos y relaciones con otros conceptos.

Según World Wide Web Consortium (W3C), extraído del Documento de requisitos de OWL[19]:

> “Una ontología define los términos a utilizar para describir y representar un área de conocimiento. Las ontologías son utilizadas por las personas, las bases de datos, y las aplicaciones que necesitan compartir un dominio de información (un

dominio es simplemente un área temática específica o un área de conocimiento, tales como medicina, bienes inmuebles, reparación automovilística, gestión financiera, etc.). Las ontologías incluyen definiciones de conceptos básicos del dominio y las relaciones entre ellos, que son útiles para los ordenadores [...]"

2.3.2. Componentes de las Ontologías

Una ontología esta compuesta por los siguientes elementos[17]:

Conceptos: son las ideas básicas que se intentan formalizar. Los conceptos pueden ser abstractos o concretos, elementales (electrón) o compuestos (átomo), reales o ficticios. Un concepto puede ser dividido en otros conceptos, los cuales representarán ideas más específicas que el concepto al que pertenece.

Relaciones: representan la interacción y enlace entre los conceptos del dominio. Suelen formar usualmente una taxonomía del dominio. Por ejemplo: subclase-de, parte-de, parte-exhaustiva-de, conectado-a, etc.

Funciones: son un tipo concreto de relación donde se identifica un elemento mediante el cálculo de una función que considera varios elementos de la ontología. Por ejemplo, pueden aparecer funciones como categorizar-clase, asignar fecha, etc.

Instancias: se utilizan para representar objetos determinados de un concepto.

Axiomas: son teoremas que declaran restricciones que deben cumplir los elementos de la ontología. Por ejemplo: "Si A y B son de la clase C, entonces A no es subclase de B", "Para todo A que cumpla la condición C1, A es B", etc.

2.3.3. Clasificación de las Ontologías

Según el grado de generalidad o nivel de dependencia a una tarea o visión en particular, Mizoguchi y colegas (Mizoguchi et al, 1995)[20] presentan la siguiente clasificación de las ontologías:

Ontologías de Alto Nivel (genéricas): describen conceptos muy generales como espacio, tiempo, materia, evento, acción, etc., los cuales son independientes de un problema

o dominio en particular. Por lo tanto, parece razonable, al menos en teoría, tener ontologías unificadas de alto nivel para grandes comunidades de usuarios.

Ontologías de Dominio y Ontologías de Tarea: describen, respectivamente, el vocabulario relacionado a un dominio genérico (como medicina o automóviles), o una tarea o actividad genérica (diagnóstico o venta), mediante la especialización de los términos introducidos en la ontología de alto nivel.

Ontologías de Aplicación: describen conceptos que dependen tanto de un dominio como de una tarea en particular, los cuales frecuentemente son especializaciones de ambas ontologías. A menudo, estos conceptos corresponden a los roles desempeñados por entidades del dominio mientras realizan cierta actividad. Contienen conocimiento esencial y especifico, para modelar una aplicación particular bajo consideración.

Otra clasificación de ontología propuesta por Van Heist (Van Heist et al.,1997)[21], es la siguiente:

Ontologías Terminológicas: especifican los términos que son usados para representar el conocimiento en el universo del discurso. Suelen ser usadas para unificar vocabularios en un campo determinado.

Ontologías de Información: especifican la estructura de almacenamiento de bases de datos. Ofrecen un marco para el almacenamiento estandarizado de información.

Ontologías de Modelado de Conocimiento: describen la información usando conceptualizaciones con una estructura interna, mucho más rica que las ontologías de información. Este tipo de ontologías suelen representar el conocimiento teniendo en cuenta el uso que se le va a dar.

2.3.4. Arquitectura Orientada por Ontologías (ODA)

2.3.4.1. Ingeniería dirigida por Modelos

La Ingeniería Dirigida por Modelos (Model Driven Engineering o MDE) es una metodología de desarrollo de software que se centra en la creación de modelos o abstracciones, para modelar y especificar sistemas (computacionales, reales, etc.)

El primero en definir la estructura conceptual fue Stuart Kent[22], cuando "abordo el escurridizo problema del desarrollo de sistemas, promoviendo el uso de los modelos como primer artefacto para su construcción y su mantenimiento". Sin embargo, el artefacto realmente novedoso definido por MDE son las transformaciones, que permiten establecer un mapeo trazable desde los modelos definidos hasta la implementación final.

La arquitectura dirigida por modelos (Model Driven Architecture o MDA)[23] es tal vez la más prometedora y ambiciosa propuesta de la Object Management Group (OMG), se basa en los conceptos de MDE pero utilizando los estándares de la OMG como son: MOF, XML, OCL, UML y CWM. En MDA se propone una forma de desarrollo del software basada en varios niveles de abstracción. Los niveles de abstracción van de lo mas abstracto a lo mas concreto, comenzando su ciclo de vida por la capa CIM y terminando en el Código. En la figura 2.13 se pueden observar las tres capas del MDA, acompañadas con los diferentes proceso y diagramas UML a usar en cada respectiva capa.

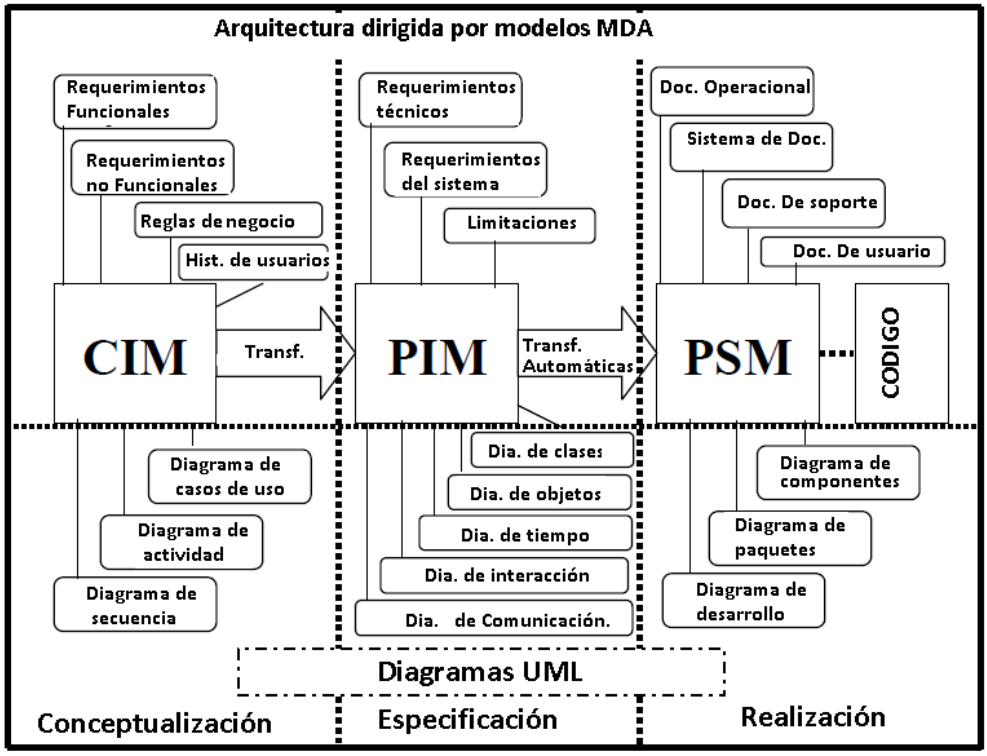

Figura 2.13: Ciclo de Vida del MDA

Los niveles o modelos que componen el ciclo de vida del MDA son los siguientes:

Modelo Independiente de la Computación (CIM): se centra en los requerimientos, y representa el nivel más alto del modelo de negocios. Se analizan los estándares y se describen aquellas interacciones que tienen lugar entre los procesos, y las responsabilidades de cada trabajador, sea o no humano. Un objetivo fundamental del CIM es que cualquiera pueda entender el negocio y los procesos del mismo.

Modelo Independiente de la Plataforma (PIM): Se modelan los procesos y estructuras del sistema, sin hacer ninguna referencia a la plataforma en la (o las) que será desplegada la aplicación. A su vez, ignora los sistemas operativos, los lenguajes de programación, el hardware y la topología de red. Suele ser el punto de entrada de todas las herramientas para MDA, dejando de lado el CIM.

Modelo Específico de la Plataforma (PSM): Representa la proyección de los PIMs en una plataforma específica. Un PIM puede generar múltiples PSMs, cada uno para una tecnología distinta. Generalmente, los PSMs deben colaborar entre sí para una solución completa y consistente. Adicionalmente, los PSM tienen que lidiar explícitamente con los sistemas operativos, los lenguajes de programación y las plataformas (J2EE, CORBA, .Net, entre otros).

Mientras el MDA proporciona un marco de gran alcance y probada en los Sistemas y la Ingeniería del Software, las tecnologías semántica, naturalmente, pueden ampliarla para permitir la representación de vocabularios inequívocos de dominio, la verificación del modelo de consistencia, validación, y el incremento de las nuevas capacidades de expresividad en la representación de restricciones. Los Modelos semánticos u ontologías, amplían la metodología base de la OMG, dando lugar a la Arquitectura Dirigida por Ontología.

2.3.4.2. Ingeniería dirigida por Ontologías

La arquitectura dirigida por Ontología (Ontology-Driven Architecture o ODA)[24] se basa en ontología. Una ontología representa un modelo abstracto que identifica los componentes más relevantes de cierto fenómeno del mundo real, y los describe en un lenguaje formal, es decir, que puede ser entendido por una computadora, a través de conceptos, propiedades, relaciones, funciones, restricciones y axiomas. Una de las tendencias actuales en la ingeniería de software es el uso de ontologías como parte del proceso del desarrollo dirigido por modelos,

es decir, como medio para representar los modelos conceptuales que definen al sistema a desarrollar en el paradigma MDA.

La ventaja principal del uso de ontologías como modelo en MDA es que en una sola ontología es posible representar la estructura del sistema, su comportamiento y el modelo de datos, además de describir la lógica de negocio y restricciones de software en el mismo modelo.

Entre otras ventajas del uso de las ontologías, además de las que por defecto se heredan de MDA, es que proporcionan un entendimiento compartido del dominio del problema y una especificación formal de los requisitos de un sistema, lo que permite el rehúso de definiciones existentes en un dominio; además proporcionan mayor soporte para inferencias lógicas.

En pocas palabras, ODA centra su desarrollo en las ontologías, las cuales además de describir el dominio, son capaces de representar el contenido semántico del sistema. En cambio, el MDA se centra el desarrollo en los modelos, posando en ellos toda la carga de especificación, análisis y diseño del sistema a desarrollar. El enfoque de ODA es complementario del MDA.

2.3.4.3. Metodología para el Desarrollo de aplicaciones basado en el paradigma ODA

La metodología para el desarrollo de aplicaciones basado en el paradigma ODA[25], consiste básicamente en las tres etapas de desarrollo, conceptualización, especificación y realización, que establece el MDA. En el desarrollo de cada una de estas etapas, se definen una serie de diagramas que ayudarán a generar la ontología resultante para cada fase (ver figura 2.14).

Figura 2.14: Fases de la Metodología

Las tres fases del desarrollo metodológico son las siguientes:

Conceptualización o Modelo independiente de la computación (CIM): En esta fase se representa de manera compacta la conceptualización del problema a través de un modelo ontológico; que usa como insumo los siguientes diagramas UML: Casos de Uso, Actividades y Secuencia. Ese modelo ontológico podría ser extendido por ontologías propias del dominio de aplicación, las cuales pueden describir estándares, normas, etc., lo que enriquece la descripción semántica de la conceptualización del problema. A ese modelo ontológico se le denomina *modelo ontológico conceptual del problema*.

Especificación o Modelo independiente de la plataforma (PIM): En esta fase el objetivo fundamental es definir una ontología centrada en el diseño del sistema, sin considerar los detalles necesarios para su implementación en una plataforma determinada. Además, se debe definir la ontología que describe los paradigmas de arquitectura de software a usar para el desarrollo de la aplicación. De esta manera, la ontología de la

capa PIM fusiona los aspectos específicos de diseño del sistema bajo desarrollo con los paradigmas de software a usar en su desarrollo. El diagrama UML fundamental para esta fase es el Diagrama de Clases, y la ontología producto de esta fase es el *modelo ontológico de diseño del sistema.*

Realización o Modelo específico de la plataforma (PSM): En esta fase se especifica detalladamente en una ontología el proceso de implementación del sistema bajo desarrollo, así como sus componentes, y la interacción entre ellos. Para ello, se usan como insumo los diagramas UML de Componentes y de Paquetes. Esa ontología es el *modelo ontológico de implantación del sistema.*

2.4. Contenidos Digitales

2.4.1. Definición de Contenidos Digitales

El Ministerio de Tecnologías de la Información y las Comunicaciones (MinTIC) de Colombia lo define como: "Los contenidos digitales son cualquier forma de datos o información en forma digital (es decir, un archivo electrónico)"[26].

Cualquier archivo electrónico como: fotos, artes gráficas, documentos de investigación, artículos, informes, estadísticas, bases de datos, planes de negocios, diseños de ingeniería, e-libros, multimedia (música y películas), objetos de aprendizajes (OA), cursos en linea, entre otros, son considerados Contenidos Digitales.

La gran ventaja que suponen los contenidos digitales es la facilidad de acceso a multitud de información o aplicaciones, sin barreras geográficas y ahorro temporal. Gracias a Internet y los buscadores (tipo Google, Yahoo, Ask, etc), introducimos algunos metadatos o términos, y podremos localizar los recursos que necesitamos.

Por otro lado, otro beneficio de este tipo de formatos es que su edición no supone unos gastos elevados: teniendo un ordenador, una impresora y los programas informáticos apropiados, se pueden sacar informes, fotos, composiciones musicales, etc. De hecho, la mayoría de las empresas y administraciones públicas han realizado un esfuerzo para convertir a formato digital gran cantidad de la documentación que han ido generando en soporte físico (por ejemplo, papel), a lo largo de su historia.

2.4.2. Clasificación de los Contenidos Digitales

Los contenidos digitales pueden ser clasificados en una variedad de maneras, por ejemplo, Nájera (2004)[27] propone la atención a diversos preceptos, entre los que destaca el de la función que desempeñan los contenidos, traducida en criterios de interés de uso, aplicación, y de producción, de los que resultan las tres categorías siguientes:

Informativos: Son contenidos compuesto por información de utilidad y que explican o aclaran desde una teoría hasta datos específicos y concretos, pasando por conceptos claves. Su marco de referencia es temático. Ejemplos son: Enciclopedias, tratados, informes, artículos, noticias.

Apoyo educativo: Es una gama que incluye información que favorece la incorporación de conocimientos, pero que no tiene implícitamente una labor de instrucción. Ejemplos son: Guías de estudio, manuales, apuntes, problemarios.

Tratamiento educativo: Son contenidos que median el acceso a la información para procurar un aprendizaje formativo o informativo con un lenguaje apropiado, según el usuario al que va dirigido. Ejemplos son: tutoriales, objetos de aprendizajes, cursos en línea

Otra clasificación interesante de los contenidos digitales es según su distribución[28]:

Audio: Es un sonido o secuencia de sonidos que ha sido convertido a un formato binario para poder ser almacenado en un ordenador. Por ejemplo: audio streaming, podcast, audiolibros, entre otros.

Imagen: Son contenidos formados por píxeles, los cuales pueden ser generados por copia del entorno físico (escaneado, fotografía digital), o generados digitalmente a través de aplicaciones (gimp, inkscape).

Vídeo: Es una mezcla de imagen y audio en un sólo archivo digital. Por Ejemplo: flash, videos streaming (youtube), videoconferencia, etc.

Texto: Es una composición de signos codificados en un sistema de escritura (como un alfabeto) que forman una unidad, almacenado de manera digital (ebooks, bibliotecas digitales, blogs).

Otros: Es cualquier contenido digital que no puede ser agrupado en las categorías anteriores, por ejemplo: vídeo juegos, aplicaciones, entre otros.

2.4.3. Ontology for media resources 1.0 de la W3C

Un estándar ontológico para la representación de Contenidos Digitales es la ONTOLOGY FOR MEDIA RESOURCES 1.0 de la W3C[29] (ver figura 2.15), su objetivo es proveer un conjunto de propiedades estándares, tales como: **locator** para la URL del contenido, **title** para el titulo del contenido, **description** para la descripción del contenido, **agent** para la información del autor/editorial, entre otros más. Asimismo, también busca proporcionar la información básica necesaria para la interoperabilidad entre los distintos tipos de formatos de metadatos y contenedores de medios disponibles en la Internet.

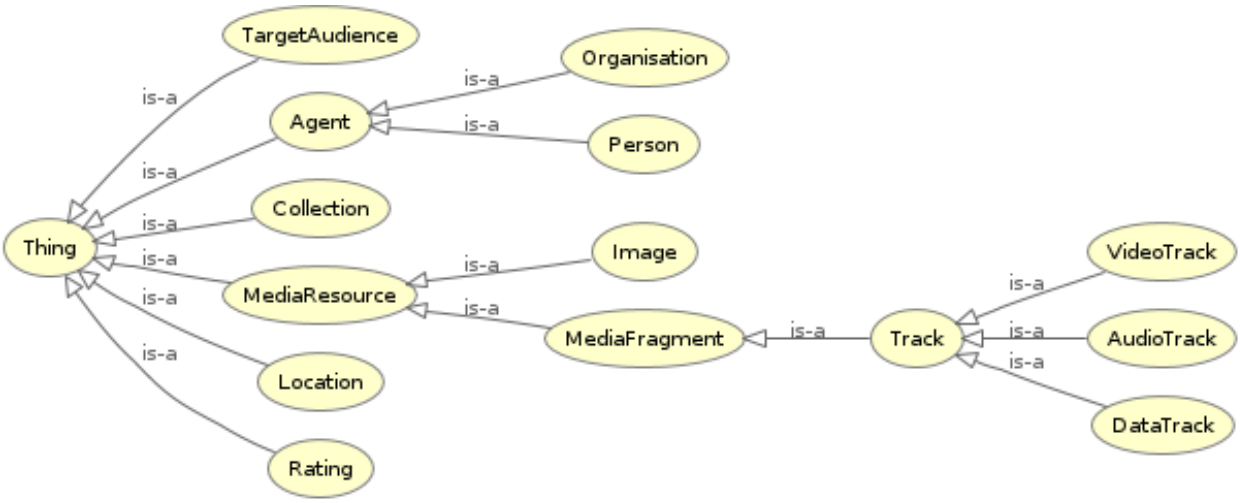

Figura 2.15: Ontology for media resources 1.0

Las propiedades de Ontology for Media Resources se pueden dividir en las siguientes categorías: Información de identificación del recurso, Información sobre su creación, Descripción sobre el contenido, Información relacional, Derechos sobre el recurso, Información sobre su distribución, e Información técnica.

Las características más destacadas de este estándar son:

- Proporciona una lista muy completa de mapeos globales desde varios tipos de formatos multimedia hacia la propia ontología, entre estos formatos se encuentra: Cablelabs 1.1,

DIG35, Dublin Core, EBUCore, EXIF, ID3, IPTC, LOM, Media RSS, MPEG-7, OGG, QuickTime, DMS-1, TTML, TV-Anytime, TXFeed, XMP y Protocolo del API de datos de YouTube.

- Posee la capacidad de referenciar cualquier contenido que este disponible en la Internet, permitiendo su localización a través de mecanismos semánticos.
- Es un estándar de la W3C con apoyo de una gran comunidad, donde están involucradas grandes empresas (Apple, Samsung, Nokia) y universidades de diferentes países (Universidad Politécnica de Madrid, Universidad de Amsterdam, etc).

Esta ontología es especialmente útil cuando se requiere describir recursos multimedia publicados en la web que poseen distintos formatos de metadata. La especificación completa del vocabulario y los mapeos se puede encontrar en su página web[29].

2.4.4. Massive Open Online Course (Mooc)

Finalmente, un contenido digital de especial interés en este trabajo serán los MOOC, termino acuñado por Dave Cormier y Bryan Alexander en el año 2008. Según el artículo What is the theory that underpins our moocs? de George Siemens[30], proviene del acrónimo en ingles Massive Open Online Course. El diccionario de Oxford lo define como: "un curso de estudio a disposición en Internet de forma gratuita a un número muy grande de personas".

Un curso puede ser considerado MOOC si cumple los siguientes requisitos:

Massive: El número de posibles matriculados es, en principio, ilimitado, o bien en una cantidad muy superior a la que podría contarse en un curso presencial. El alcance es global.

Open: Los materiales son accesibles de forma gratuita en Internet. Ello no implica que puedan ser reutilizados en otros cursos.

Online: El curso es a distancia, e Internet es el principal medio de comunicación. No requiere la asistencia a un aula.

Course: Debe contar con una estructura orientada al aprendizaje, que suele conllevar una serie de pruebas o de evaluaciones para acreditar el conocimiento adquirido.

Algunas universidades de talla mundial, tales como Harvard y el MIT, ya están incursionando en los MOOC. MIT comenzó hace algunos años montando en línea y dando acceso gratuito al material de todos los cursos que ofrece. Desde hace unos años existe una alianza de las dos universidades para ofrecer varios MOOC, mediante la propuesta edX [31].

Las empresas MOOC Coursera y Udacity fueron creados por Daphne Koller, Andrew Ng, Sebastian Thrun y antiguos docentes, todos de la universidad de Stanford, en California [32].

Existen propuestas independientes, con cursos gratuitos y pagos, como Udemy y Canvas Network. A las anteriores, se une MiriadaX [33], iniciativa española promovidas por el Banco Santander, Universia y Telefónica, que ofrece cursos montados por las Universidades: Politécnica de Madrid, de Huelva, Politécnica de Valencia, UAPA y Pompeu Fabra, entre otras.

Capítulo III

Diseño del Sistema de Gestión de Contenidos Digitales

A continuación, se establecerá el diseño de la aplicación en dos pasos: en primer lugar y con un mayor nivel de abstracción, se mostrará el diseño de la arquitectura de forma que quede definido el comportamiento global del sistema, y en segundo lugar, se profundizará en la estructura de la plataforma para los contenidos digitales.

3.1. Arquitectura de Base del Proyecto Madre

El diseño de la arquitectura base del Proyecto Madre esta apoyada sobre dos pilares primordiales. Primero, el modelo filosófico de las nubes descrito en el marco teórico, y segundo, la Plataforma Web esta basada en la Arquitectura Orientada a Servicio (SOA). En la figura 3.1 se observan tres capas de gestión, donde cada capa administra una nube del modelo filosófico, la capa de **Gestión Administrativa** es la encargada de manejar la nube de Auto-Formación, la capa de **Gestión Objetos Virtuales** se encarga de la nube de Fuentes de Conocimiento, y por ultimo la capa de **Gestión Paradigma Aprendizaje** que controla la nube de Paradigmas de Aprendizajes. Cada capa tiene sus propias estructuras de almacenamiento, que son ontologías específicas a cada dominio de cada capa, cuyas instancias representan la información guardada. A eso se llama en la figura 3.1 **Base de Datos Extendida**, la cual, además, tiene la capacidad de soportar procesos de razonamiento sobre ella. En este trabajo se desarrolló la plataforma para la gestión de la nube de fuentes de conocimiento compuesta por Contenidos Digitales, que es parte de la segunda capa, el resto de capas fue, o está siendo

desarrollada, en otros trabajos: la plataforma de gestión de la nube de auto-formación[3], la plataforma para la gestión de la nube de fuentes de conocimiento compuesta por objetos de aprendizaje[2], y la plataforma de gestión de la nube de paradigmas de aprendizajes[1].

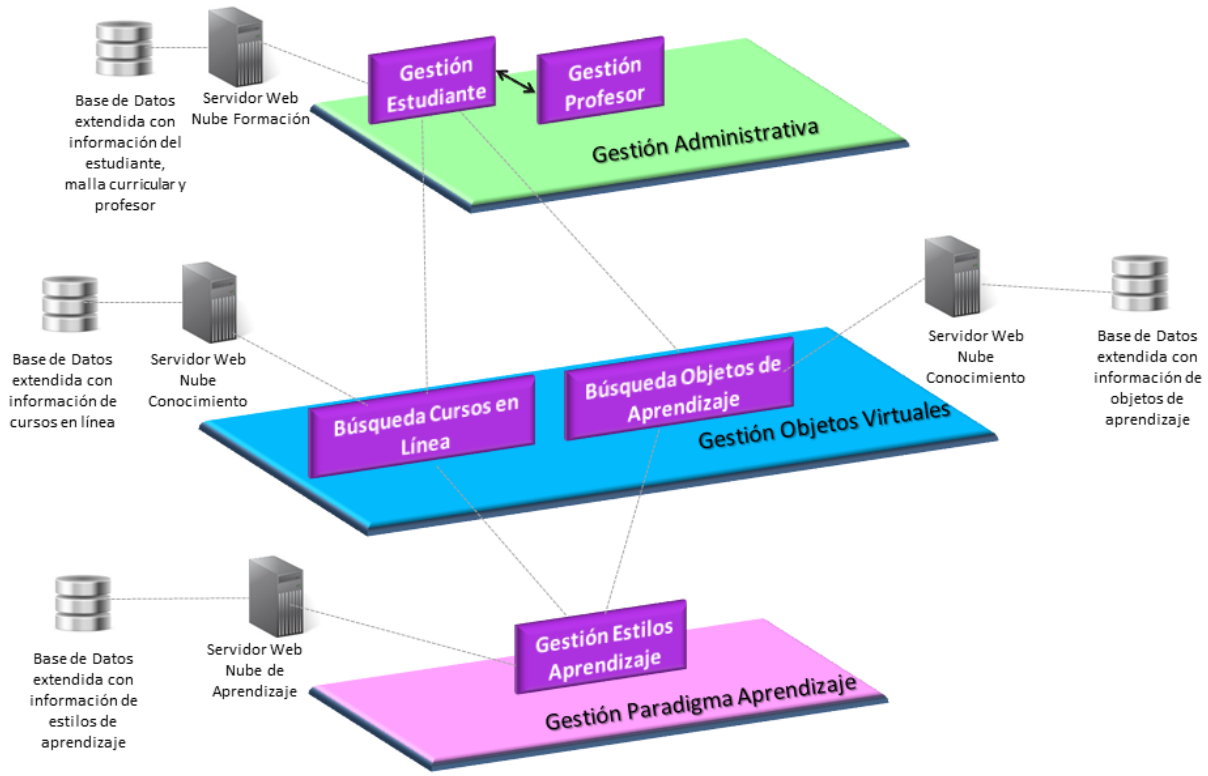

Figura 3.1: Arquitectura Funcional de la Plataforma Web[3]

Un ejemplo de un posible proceso de invocación entre las nubes se muestra en la figura 3.2, para el caso de un estudiante que esté tomando un curso (modulo) de la malla curricular. Se inicia con la solicitud por parte de la nube de auto-formación (un estudiante ya registrado) a la nube de fuentes de conocimiento de un listado de objetos de aprendizaje o contenidos digitales para un tema en particular correspondiente al módulo y al perfil del aprendizaje del estudiante (paso 1). Ese listado debe ser compatible con la descripción del estudiante; para ello, la nube de fuentes de conocimiento realiza una consulta a la nube de paradigmas de aprendizajes de cuales herramientas, actividades, mecanismos de evaluación y tipo de obras, debe usar un estudiantes con ese perfil educativo (paso 2). Esa información es enviada por

la nube de paradigmas de aprendizajes a la nube de auto-formación (paso 3). Una vez que se conoce el conjunto de características de los objetos virtuales a buscar, la nube de fuentes de conocimiento se encarga de buscar (en su base de datos de objetos certificados o en internet) los objetos virtuales adecuados, y le entrega a la nube de auto-formación un listado ordenado por prioridad de los distintos objetos de aprendizaje o contenidos digitales asociados a ese estudiante en particular (paso 4).

Cada componente de las nubes pueden estar instalado en uno o varios servidores web con el conjunto de servicios a proveer. El o los servicios se conectan a la información que ha sido almacenada en las bases de datos extendidas (u ontologías) respectivas, para poder responder a las solicitudes de otras nubes.

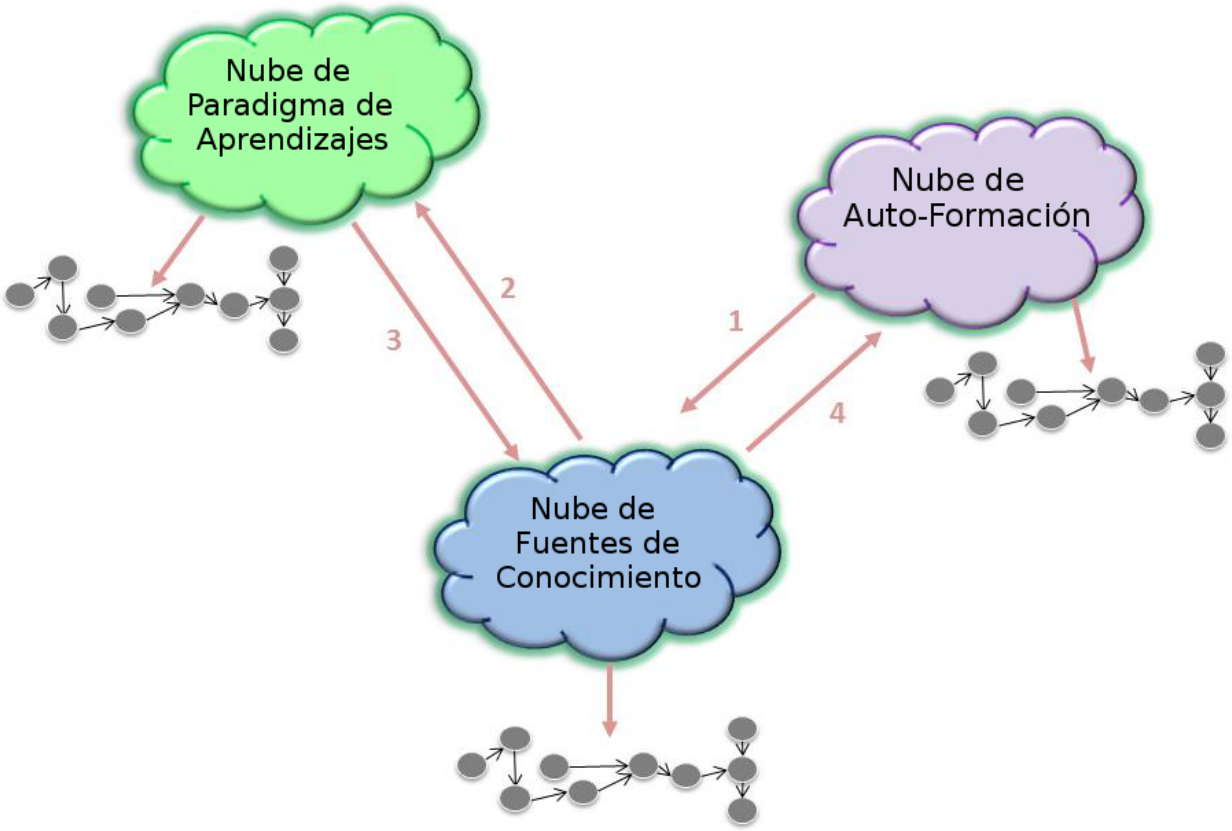

Figura 3.2: Proceso de invocación entre las nubes

3.2. Fase de Diseño

Esta fase está compuesta por el Modelo Independiente de la Computación (CIM) que recoge los requerimientos del sistema, y por el Modelo Independiente de la Plataforma (PIM) que detalla la arquitectura funcional, y los procesos y estructuras que lo componen.

3.2.1. Conceptualización o Modelo Independiente de la Computación (CIM)

En este modelo se comienza por definir los requisitos funcionales y no funcionales de la plataforma, y culmina con la ontología conceptual de la plataforma web.

3.2.1.1. Requisitos funcionales

El conjunto de entradas, comportamientos y salidas que requiere el sistema, tanto para el cliente (sistemas externos) como para el especialista, que son los actores fundamentales, se describen a continuación:

Consultar: Permite al estudiante y al especialista consultar la información de los contenidos digitales, según la necesidad que se tenga, por ejemplo, título, descripción, herramientas, actividades, entre otros. Asimismo, también debe permitir realizar consultas de los contenidos digitales que coincidan con un estilo de aprendizaje y tema específico, ya que es una de las necesidades primordiales de los clientes.

Registrar: Esta funcionalidad permite al especialista agregar los contenidos digitales, con la información que se tenga del contenido, por ejemplo, herramientas, actividades, instrumentos, titulo, entre otros.

Eliminar: Permite al especialista borrar contenidos digitales que se consideren que no cumplen con las necesidades latentes.

En la tabla 3.1 se detalla un ejemplo del diseño de caso de uso de uno de los requerimientos funcionales, el resto está en el apéndice A:

Caso de Uso: Consultar	
Código	CU-01
Descripción	El sistema debe consultar en la ontología la información de los Contenidos Digitales, como título, localizador, herramientas, actividades, instrumentos, entre otros.
Actores	Cliente/Especialista
Pre-Condición	Conocer el identificador del Contenido Digital o el tópico y estilo de aprendizaje
Condición de Éxito	Información encontrada en la ontología o en internet
Condición de Fracaso	No se encontro la información en la ontología o en internet
Flujo Normal	1.-El cliente o especialista invoca al método correspondiente 2.-El método comprueba los parametros y recibe la solicitud 3.-Busca el o los contenido en la ontología o en internet 3.-Retorna el o los contenidos
Flujo Alternativo	2.A.-Si los parametros no estan correctos retorna error

Tabla 3.1: Caso de uso "Consultar"

En la figura 3.3 se define el Diagrama de Caso de Uso general para los contenido digitales, donde se puede observar los actores del sistema con su funcionalidad, descrita a continuación:

Cliente: solo puede consultar la información de los contenidos digitales a través de los distintos métodos del caso de uso "Consultar".

Especialista: tiene acceso total al sistema, puede consultar como lo hace el cliente, pero adicionalmente puede eliminar contenidos digitales y registrar nuevos contenidos digitales al sistema.

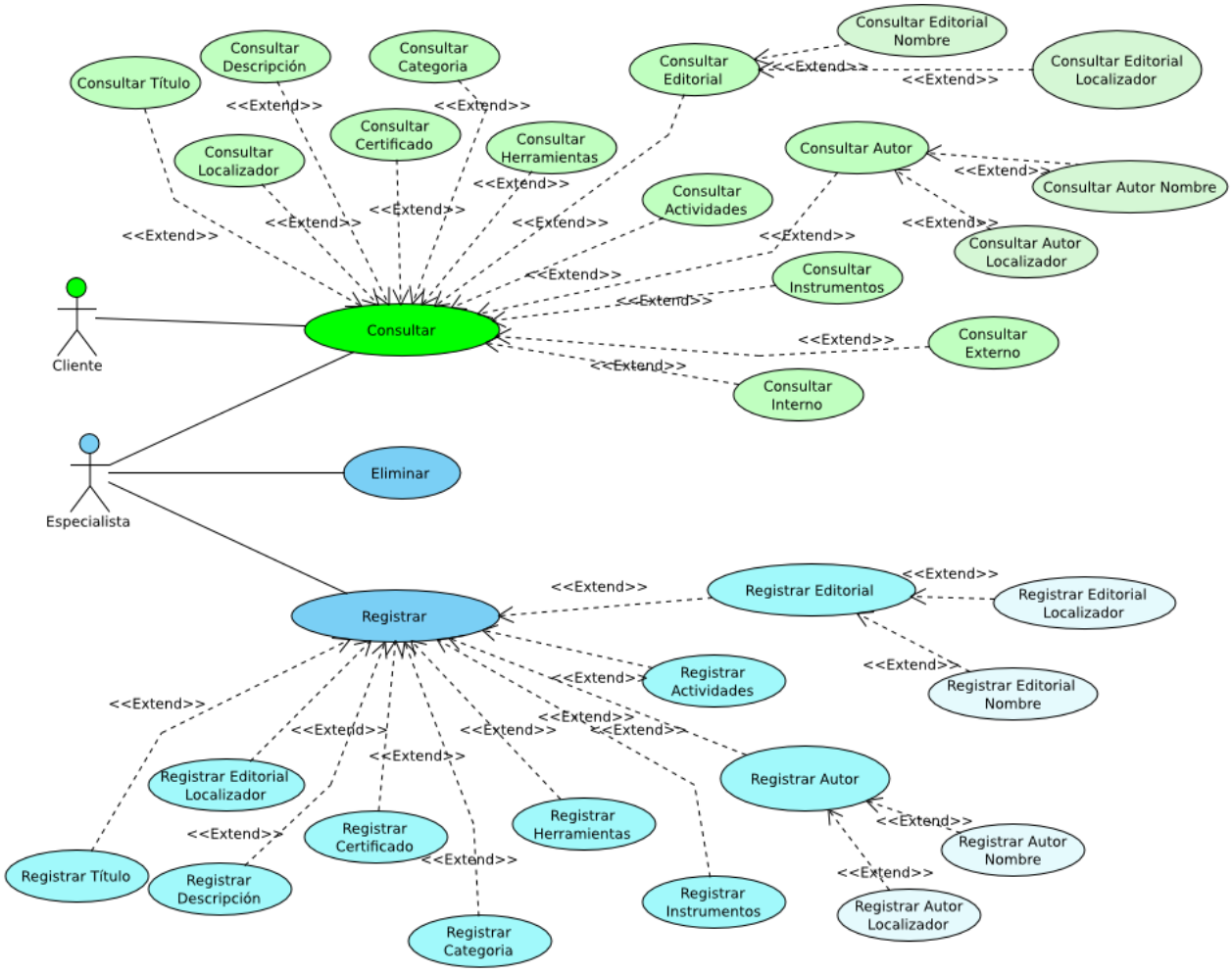

Figura 3.3: Diagrama de Casos de Uso general para Los Contenidos Digitales

Los procesos de consultar, registrar y eliminar en general son bastante sencillos, todos parten del mismo principio, con solo tener el **"id"** del contendido digital y la necesidad específica, se puede solicitar información a través de **Consultar**, guardar información con **Registrar**, y borrar información con **Eliminar**. Hay un caso particular que es interesante describir con los diagramas de actividades, y es cuando se necesita consultar contenidos digitales para un tópico específico dependiendo del estilo de aprendizaje, para esto se cuenta con dos métodos: Consultar Interno y Consultar Externo.

Antes de describir el funcionamiento de los dos métodos, hay que explicar que es un contenido digital certificado y uno no certificado. Se dice que un contenido es certificado, sí el

mismo ha sido revisado y categorizado por un especialista, indicando que es un contenido acto para fortalecer o clarificar un determinado tema, y los contenidos que no pasan el proceso se considera que son contenidos digitales no certificados.

El método Consultar Interno tiene la finalidad de obtener información alojado en la ontología del sistema sobre contenidos públicos en la web, estos contenidos son certificados. En cambio, el método Consultar Externo obtiene la información sobre los contenidos en buscadores propios de las distintas plataformas en internet (por ejemplo, Coursera, Udacity, Miranda X, entre otros), y estos contenidos son no certificados. La figura 3.4 muestra de manera general el proceso de ambos métodos.

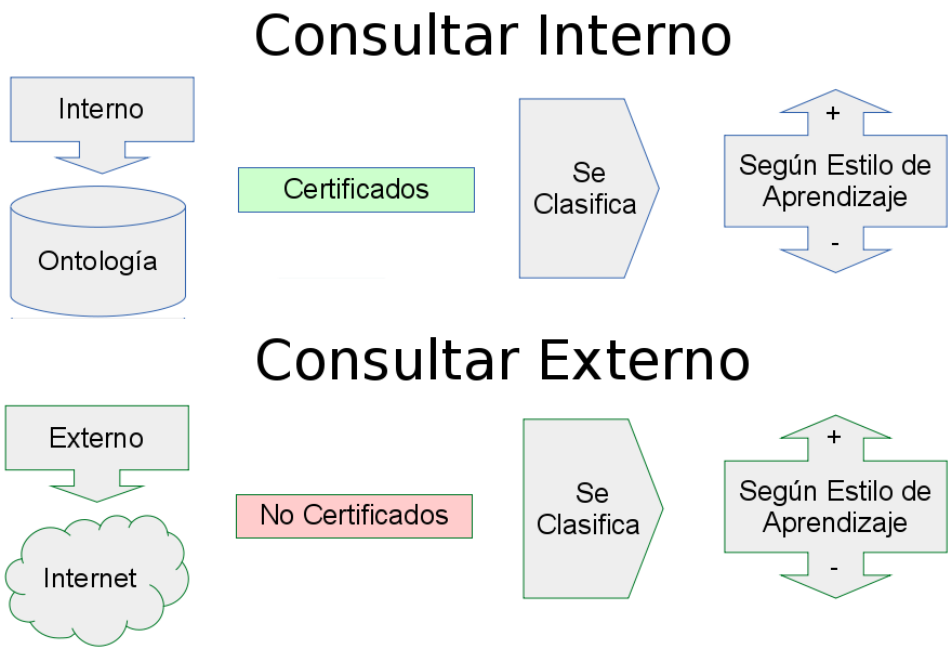

Figura 3.4: Métodos Consultar Interno y Externo

A continuación se desarrollan los diagramas de actividades para Consultar Interno (figura 3.5) y Consultar Externa (figura 3.6).

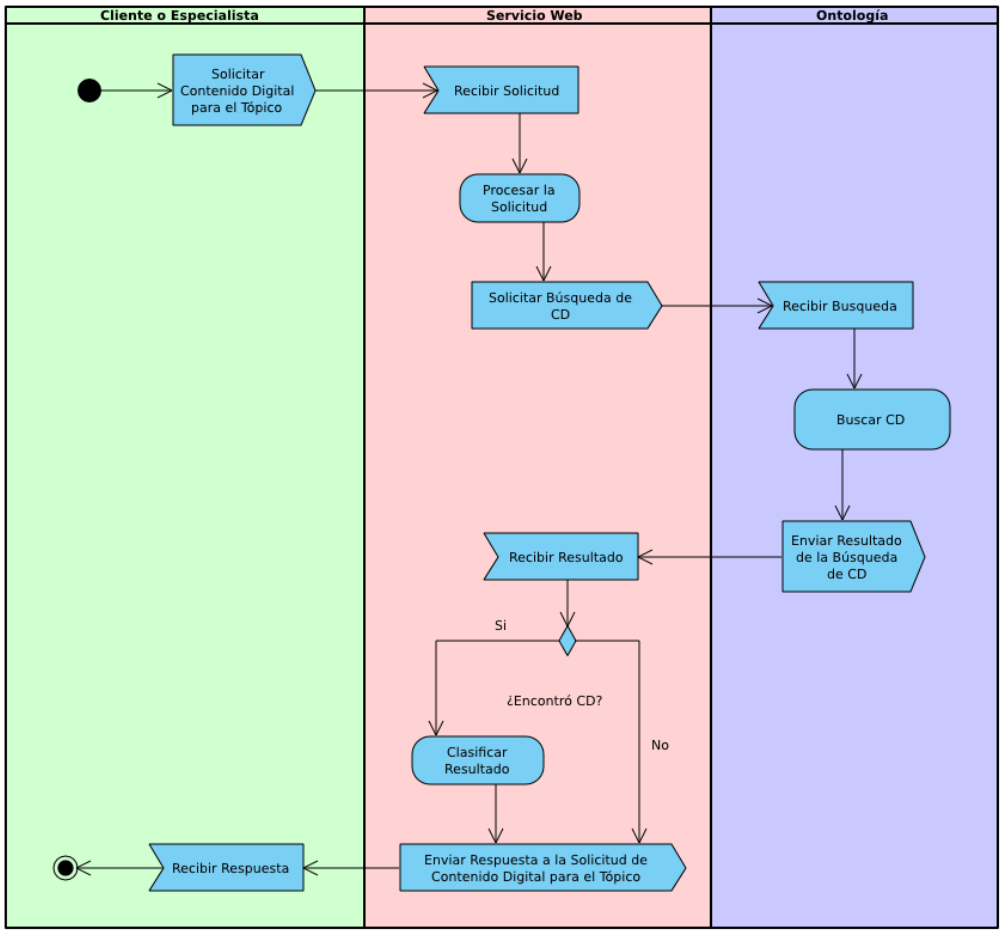

Figura 3.5: Diagrama de Actividades del proceso de Consultar Interno

El proceso de la figura 3.5, comienza por la solicitud de parte del cliente o especialista, indicando el tópico y el estilo de aprendizaje, el servicio lo recibe, procesa la solicitud y busca los contenidos digitales en la ontología que coincidan con el tópico. Una vez obtenido el resultado de la ontología, se clasifican los contenidos digitales certificados usando las herramientas, actividades e instrumentos del estilo de aprendizaje requerido. Así, coloca primero los contenidos digitales certificados ordenados de mayor a menor coincidencia con las herramientas, actividades e instrumentos, según el estilo de aprendizaje del estudiante; después, retorna el listado de los contenidos digitales.

La diferencia en el proceso consultar externo con respecto a consultar interno, es que los contenidos son buscados en la internet, y que se realiza una búsqueda en internet por cada fuente externa que cuente el sistema, como se puede observar en la figura 3.6.

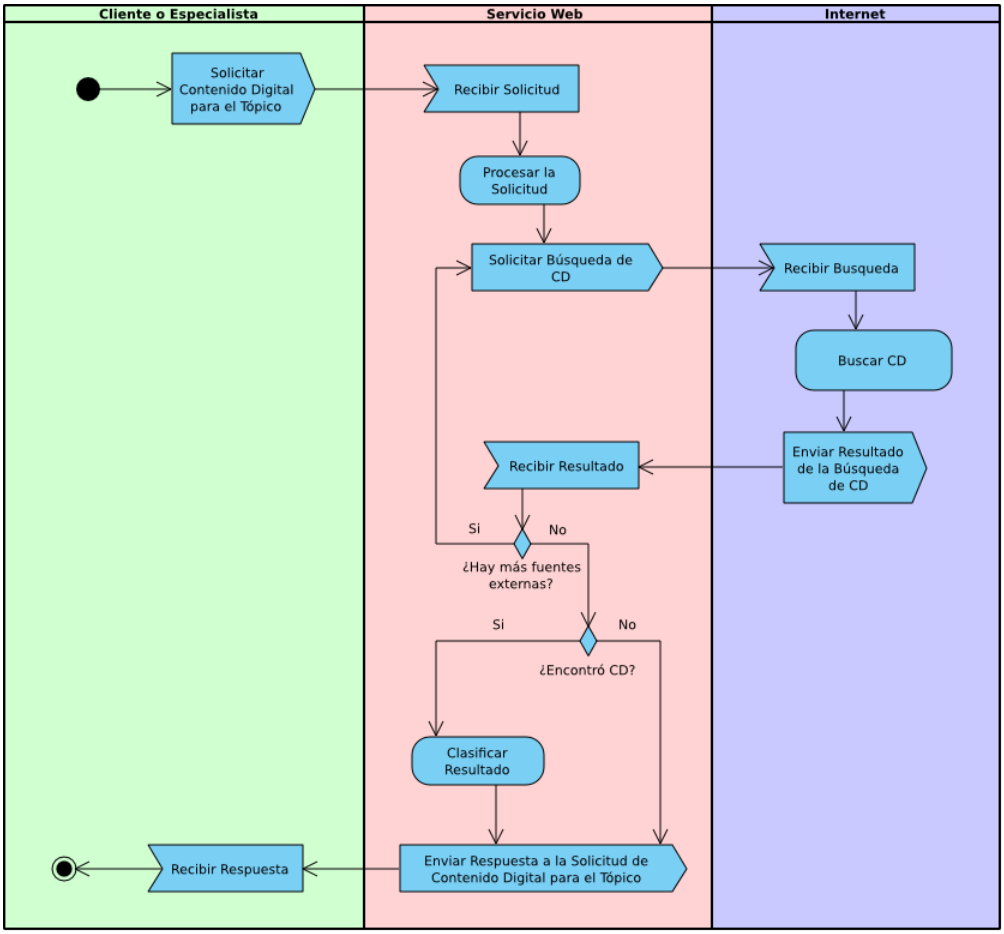

Figura 3.6: Diagrama de Actividades del proceso de Consultar Externo

Es importante destacar que cada método es independiente uno del otro, es decir, el cliente puede invocar cualquiera de los dos según la necesidad que tenga. Sin embargo, la manera ideal es llamar al método Consultar Interno, y sí este no retorna nada (no encontró contenidos) se llama al método Consultar Externo. El sistema no realiza las dos llamadas automáticamente, y la razón de esto es que los dos métodos devuelven los contenidos de distinta forma. Por ejemplo, Consultar Interno devuelve una lista de id de los contenidos digitales, y luego con ese id se puede solicitar la información que se requiera de ese contenido, ya que esta almacenado en la ontología, en cambio, Consultar Externo devuelve una lista de los contenidos con toda la información encontrada en formato de intercambio estándar JSON.

3.2.1.2. Requisitos no funcionales

Entre los requisitos no funcionales que el sistema debe cumplir, se encuentra los siguientes:

Escalabilidad: Capacidad de la plataforma para permitir incorporar nuevos módulos sin afectar su funcionamiento

Usabilidad: Interfaz sencilla, práctica y útil para los usuarios de la plataforma.

Confiabilidad: Capacidad de la plataforma para funcionar sin que se produzcan errores, y si ocurren, manejarlos sin perjudicar los demás componentes que lo integra.

Seguridad: Capacidad de la plataforma para limitar el acceso a algunas zonas según el tipo de usuario.

Interoperabilidad: Capacidad de la plataforma para intercambiar información con otros sistemas.

3.2.1.3. Ontología Conceptual

Tal y como lo indica la metodología definida en el capítulo anterior, una vez desarrollado los diagramas UML que nos interesan de la capa CIM, se culmina con el diseño de la ontología de dicha capa. En la figura 3.7 se observa la ontología de la capa CIM, definida paso a paso según la metodología methontology:

1. Construcción del glosario de términos: Se definen todos los términos nombrados en los diagramas de casos de uso y de actividades general de la plataforma.

Glosario de términos de la ontología CIM		
Nombre	**Descripción**	**Tipo**
Plataforma Web	Sistema Web que gestiona la nube de conocimiento especificamente los Contenidos Digitales	Concepto
Requerimientos funcionales	Describen la interacción entre el sistema y su ambiente independientemente de su implementación	Concepto
Requerimientos no funcionales	Describen aspectos del sistema que son visibles por el usuario que no incluyen una relación directa con el comportamiento funcional del sistema	Concepto
Sigue...		

Tabla 3.2: Glosario de términos de la ontología CIM (continua...)

Glosario de términos de la ontología CIM (cont.)		
Nombre	**Descripción**	**Tipo**
Consultar información	Proceso de consulta de información de los contenidos digitales	Concepto
Almacenar información	Almacenamiento de información de los contenidos digitales	Concepto
Inferir información	Proceso de inferencia de la información consultada	Concepto
Eliminar información	Proceso de eliminacion de los contenidos digitales	Concepto
Conexión con las otras nubes	Enlace de conexión con las otras nubes (Formación y Aprendizaje). Para acciones como consultas de contenidos digitales según el tópico y estilo de aprendizaje	Concepto
Usabilidad	Interfaz sencilla, práctica y útil para los usuarios de la plataforma	Concepto
Confiabilidad	Capacidad de la plataforma para funcionar sin que se produzcan errores y si ocurren, manejarlo sin perjudicar los demás componentes que lo integra	Concepto
Escalabilidad	Capacidad de la plataforma para permitir incorporar nuevos módulos sin afectar su funcionamiento	Concepto
Seguridad	Capacidad de la plataforma para limitar el acceso a algunas zonas según el tipo de usuario	Concepto
Interoperabilidad	Capacidad de la plataforma para intercambiar información con otros sistemas	Concepto
Contenido Digital	Capacidad de la plataforma para manejar información sobre distintos tipos de contenidos digitales	Concepto
Ontology for media resource 1.0 (figura 3.8)	Ontología estandar de la W3C para la representación de Contenidos Digitales	Concepto
Audio	Clasificación del contenido según su distribución	Concepto
Video	Clasificación del contenido según su distribución	Concepto
Imagen	Clasificación del contenido según su distribución	Concepto
Texto	Clasificación del contenido según su distribución	Concepto
Otros	Clasificación del contenido según su distribución	Concepto

Tabla 3.2: Glosario de términos de la ontología CIM

2. Construcción de la taxonomía de conceptos: Como esta ontología es un modelo del sistema independiente de la computación, se resume en una taxonomía de conceptos, la cual es mostrada en la figura 3.7; dicha ontología muestra las relaciones entre los diferentes conceptos vinculados a la plataforma web y sus requerimientos.

Figura 3.7: Ontología de Conceptualización de la Capa CIM de la plataforma

En la figura 3.7 se observa que los contenidos digitales usan el estándar de la W3C **Ontology for media resources 1.0** (Sección 2.4.3), para la base de datos que almacena la información sobre los contenidos digitales, más conceptos nuevos necesarios para ofrecer la capacidad de adaptar el contenido a un estilo de aprendizaje de un estudiante específico, que se muestran en la figura 3.8. A continuación se construye el glosarios de términos que definen los conceptos de la ontología de la base de datos.

Glosario de términos de la ontología de la Base de Datos		
Nombre	**Descripción**	**Tipo**
Agente (Agent)	Persona u organización que contribuye al contenido digital	Concepto
Audiencia (TargetAudience)	Información sobre el público objetivo para los que se destina el contendido digital	Concepto
Formación	Relación de un contenido con los distintos módulos y/o temas	Concepto
Sigue...		

Tabla 3.3: Glosario de términos de la ontología de la Base de Datos (continua...)

Glosario de términos de la ontología de la Base de Datos (cont.)		
Nombre	**Descripción**	**Tipo**
Contenido Digital (MediaResource)	Recurso que puede estar compuesto por uno o más fragmentos / pista	Concepto
Colección (Collection)	Cualquier grupo de contenidos digitales, por ejemplo, cursos, pdf, entre otros	Concepto
Localización (Location)	Ubicación relacionada con el contenido digital o agente	Concepto
Aprendizaje	Relación de un contenido con el estilo de aprendizaje	Concepto
Índice (Rating)	Información sobre la calificación otorgada a un contenido digital	Concepto
Persona (Person)	Persona o Autor que contribuye al contenido digital	Concepto
Organización (Organisation)	Organización o Editorial que contribuye al contenido digital	Concepto
Módulo	Relación de un contenido con los distintos modulos de la carrera	Concepto
Tema	Relación de un contenido con los distintos temas de la carrera	Concepto
Imagen (Image)	Imagen relacionada con el contenido digital o recurso de tipo imagen	Concepto
Fragmento (MediaFragment)	Parte o segmento que componen un contenido digital	Concepto
Instrumento Evaluación	Relación de un contenido con los distintos tipos de instrumentos de evaluación	Concepto
Herramienta Aprendizaje	Relación de un contenido con los distintos tipos de herramientas de aprendizaje	Concepto
Actividad Aprendizaje	Relación de un contenido con los distintos tipos de actividad de aprendizaje	Concepto
Pista (Track)	Especialización de los fragmentos que componen al contenido digital	Concepto
Pista Video (VideoTrack)	Especialización para proporcionar un enlace a propiedades de datos específicos, tales como la velocidad de los fotogramas, etc	Concepto
Pista Audio (AudioTrack)	Especialización para proporcionar enlaces a propiedades de datos específicos como frecuencia de muestreo	Concepto
Sigue...		

Tabla 3.3: Glosario de términos de la ontología de la Base de Datos (continua...)

Glosario de términos de la ontología de la Base de Datos (cont.)		
Nombre	**Descripción**	**Tipo**
Pista Dato (DataTrack)	Especialización para proporcionar enlaces a datos auxiliares, por ejemplo, subtítulos	Concepto

Tabla 3.3: Glosario de términos de la ontología de la Base de Datos

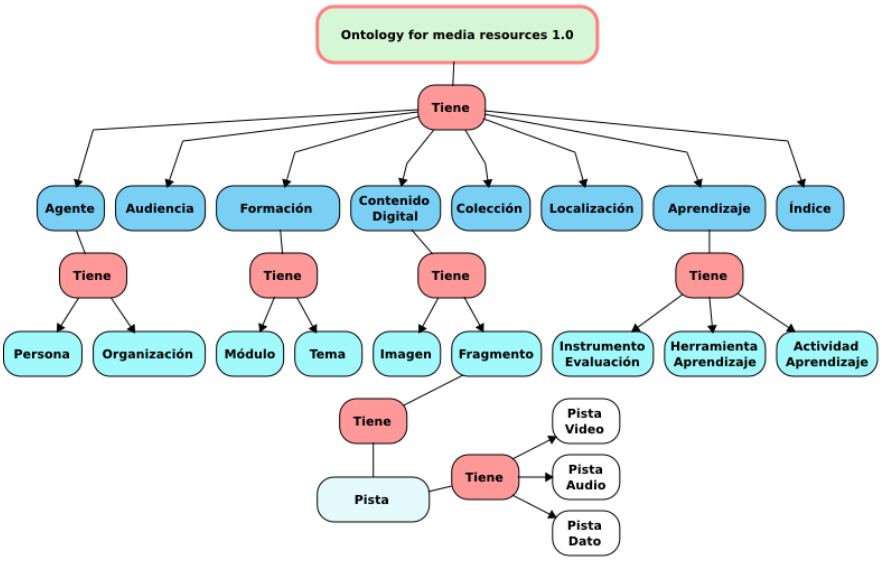

Figura 3.8: Ontology for media resources 1.0

3.2.2. Especificación o Modelo Independiente de la Plataforma (PIM)

En este modelo se describe la arquitectura funcional, sin considerar los detalles necesarios para su implementación en una plataforma determinada, acompañado de los procesos y estructuras que lo componen, para finalmente generar la ontología de diseño del sistema.

3.2.2.1. Arquitectura Funcional

La arquitectura funcional para el sistema es SOA conformada por operaciones o métodos que están disponibles en los servicios web del sistema, los cuales se muestran a continuación agrupados por funcionalidad:

Métodos de Consultar: título, localizador, certificación, descripción, categoría, herramientas, instrumentos, actividades, etiquetas, autores, nombreAutor, localizadorAutor, Edi-

toriales, nombreEditorial, localizadorEditorial, consultarInterno (consultar) y consultarExterno.

Métodos de Registrar: títuloAdd, localizadorAdd, certificaciónAdd, descripciónAdd, categoríaAdd, herramientasAdd, instrumentosAdd, actividadesAdd, etiquetasAdd, autoresAdd, nombreAutorAdd, localizadorAutorAdd, EditorialesAdd, nombreEditorialAdd, localizadorEditorialAdd.

Métodos de Eliminar: eliminar.

En la figura 3.9 se muestra el diagrama de clases general de la plataforma, donde se detalla la estructura que compone cada clase del sistema y la relación que existe entre las distintas clases. La plataforma para la gestión de la nube de fuentes de conocimiento compuesta por contenidos digitales esta conformado por las siguientes clases:

ContenidoDigital: clase principal del servicio que exhibe la estructura de los métodos.

OntologíaContenidoDigital: enlace con el servicio web **NubeAprendizajeWS** y las clases **ContenidoDigital**, **Ontología** y **Razonador**.

NubeAprendizajeWS: servicio responsable de entregar las herramientas, actividades e instrumentos para un estilo de aprendizaje específico.

Ontología: encargado de manejar la estructura de la base de datos ontológica.

Razonador: encargado de realizar las inferencias sobre la base de datos ontológica.

FContenidoDigital: servicio que permite limitar las funcionalidades de la Nube de Conocimiento a solo los métodos disponible para los clientes, brindando mayor sencillez y seguridad.

Cliente y Especialista: son las interfaces que consumen la información. Estas interfaces son servicios web, gracias a este tipo de arquitectura las interfaces pueden ser mejoradas o sustituidas sin afectar el funcionamiento de la plataforma.

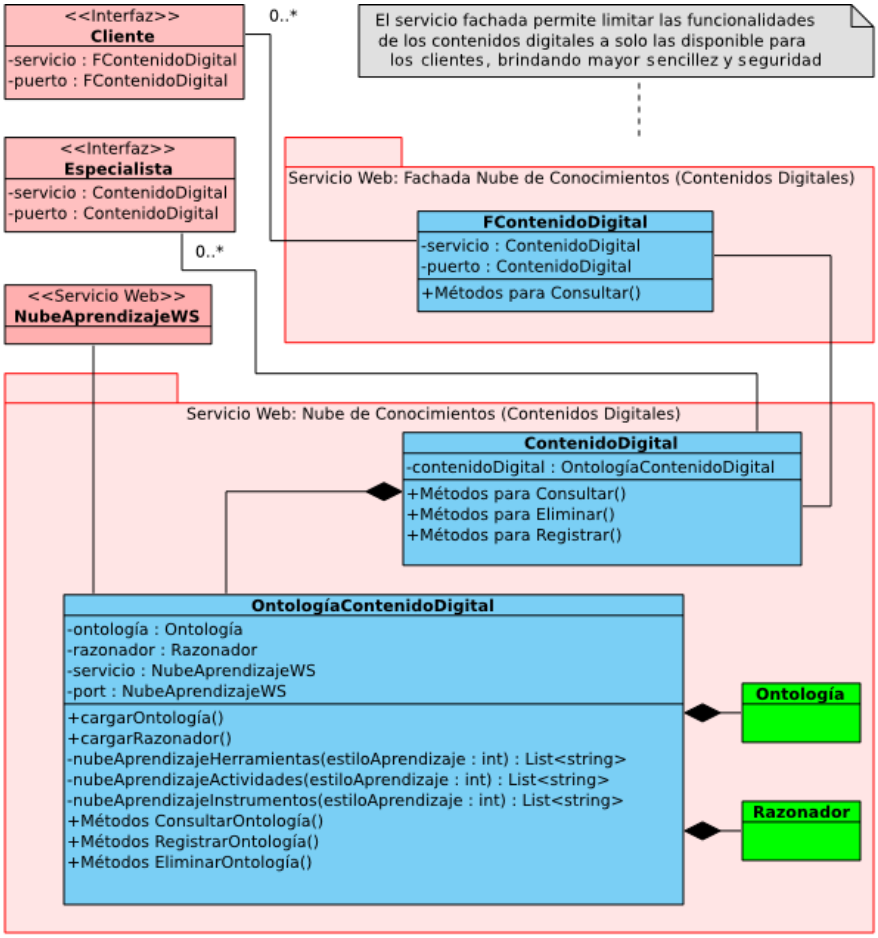

Figura 3.9: Diagrama de Clases de la plataforma

Asimismo, en esta capa se deben especificar las clases propias que caracterizan la arquitectura orientada a servicios del sistema. La figura 3.10 muestra las clases que conforman la coreografía del servicio web de la nube de fuentes de conocimiento para el caso de **ConsultarInterno** que invoca la clase principal **ContenidoDigital**, las clases que lo conforman son:

Servicio Web: representa la descripción del Servicio Web.

Operación: representa las operaciones que realiza el servicio, como por ejemplo: título, actividades, herramientasAdd, instrumentosAdd, eliminar, entre otros métodos.

Mensaje: consiste en las respuestas que emite un servicio. Para el ejemplo, los contenidos

digitales ordenados por la mayor coincidencia a su estilo de aprendizaje y que a su vez coincida con el tópico.

Función: describe cual es la función que debe cumplir ese servicio para la plataforma web. Por esta razón será invocado por el resto de servicios. Para el ejemplo, su función es permitir generar un listado de los contenidos digitales.

Restricción: describe las restricciones que tiene un servicio web, como en nuestro ejemplo, que la operación ConsultarInterno es el resultado de un proceso de inferencia sobre la ontología de contenidos digitales.

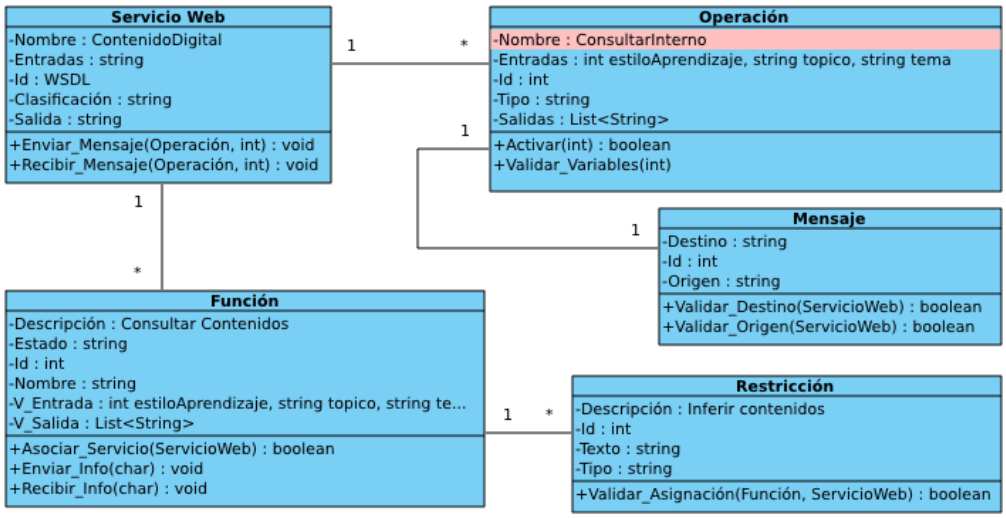

Figura 3.10: Diagrama de Clases para la operación "ConsultarInterno" del servicio web

Las propiedades que describen a las clases del servicio web para la operación ConsultarExterno es idéntico a ConsultarInterno, solo cambia el nombre de la operación.

3.2.2.2. Ontología de Diseño

En la figura 3.11 se observan las funcionalidades de la plataforma web, descritas como servicios web.

1. Construcción del glosario de términos: Se definen todos los términos nombrados en los diagramas de clases que son relevantes incorporar en la ontología.

Glosario de términos de la ontología PIM		
Nombre	**Descripción**	**Tipo**
Plataforma Web de la Nube de Fuentes de Conocimiento compuesto por Contenidos Digitales	Sistema Web que gestiona la nube de conocimiento especificamente los Contenidos Digitales	Concepto
Clientes a los Servicios Web de las Nubes de Autoformación y Aprendizaje	Representa las interfaces y servicios web que utilizan y necesitan la información de los contenidos digitales.	Concepto
Servicios Web.	Representa la descripción del Servicio Web.	Concepto
ContenidoDigital.	Servicio Web que contiene todas los métodos de consultar, registrar y eliminar.	Concepto
FContenidoDigital.	Servicio Web que limita los métodos disponible para los clientes.	Concepto
Función.	Describe cual es la función que debe cumplir ese servicio para la plataforma web.	Concepto
Operación.	Representa las operaciones que realiza el servicio.	Concepto
Restricción.	Describe las restricciones que tiene un servicio web.	Concepto
Mensaje.	Consiste en las respuestas que emite un servicio.	Concepto
Eliminar Contenidos Digitales.	Representan el conjunto de métodos para eliminar contenidos digitales.	Concepto
Registrar Contenidos Digitales.	Representan el conjunto de métodos para registrar contenidos digitales.	Concepto
Consultar Contenidos Digitales.	Representan el conjunto de métodos para consultar contenidos digitales.	Concepto

Tabla 3.4: Glosario de términos de la ontología PIM

2. Construcción de la taxonomía de conceptos

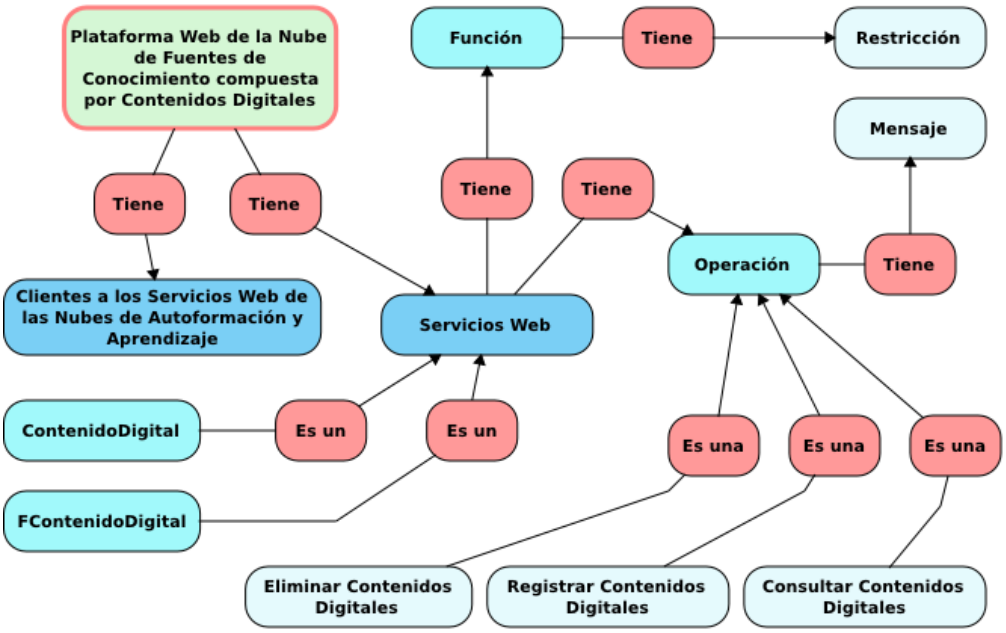

Figura 3.11: Ontología de Diseño de la Capa PIM de la plataforma

Capítulo IV

Implementación del Sistema de Gestión de Contenidos Digitales

El presente capítulo tiene por objetivo detallar la implementación del sistema planteado en el capítulo anterior. Los puntos principales a tratar son dos: la especificación del modelo PSM, y la implementación tanto del servidor como del cliente.

4.1. Arquitectura general del sistema

La arquitectura del sistema se describe en la figura 4.1, en ella se muestra una capa llamada servidor, contiene todos los proveedores de servicios web de cada una de las nubes, estos servicios web han sido desarrollados bajo el uso de la interfaz de programación de aplicaciones de Java (JAX-WS) (Sun Microsystems). Cada uno de estos servicios necesita conectarse a una base de datos extendida (ontología) que gestiona la información de cada nube. Esta conexión entre los servicios web en Java y las ontologías en protégé se realiza a través del OWL API ([CO-ODE), este gestiona la información haciendo uso del razonador Fact++ para el lenguaje OWL (W3C). Fact++ es un razonador de descripción lógica del tipo OWL DL. Este razonador permite validar la consistencia de las ontologías, así como hacer inferencia sobre la información almacenada.

La capa cliente aloja todas las referencias a los servicios web de la plataforma educativa, que permiten utilizar la información y conocimiento almacenado, además de que provee la interfaz de usuario del sistema. La comunicación entre las capas cliente y servidor se hace a través de un bus de servicios de empresa, que maneja los mensajes de enlace entre las 2

capas.

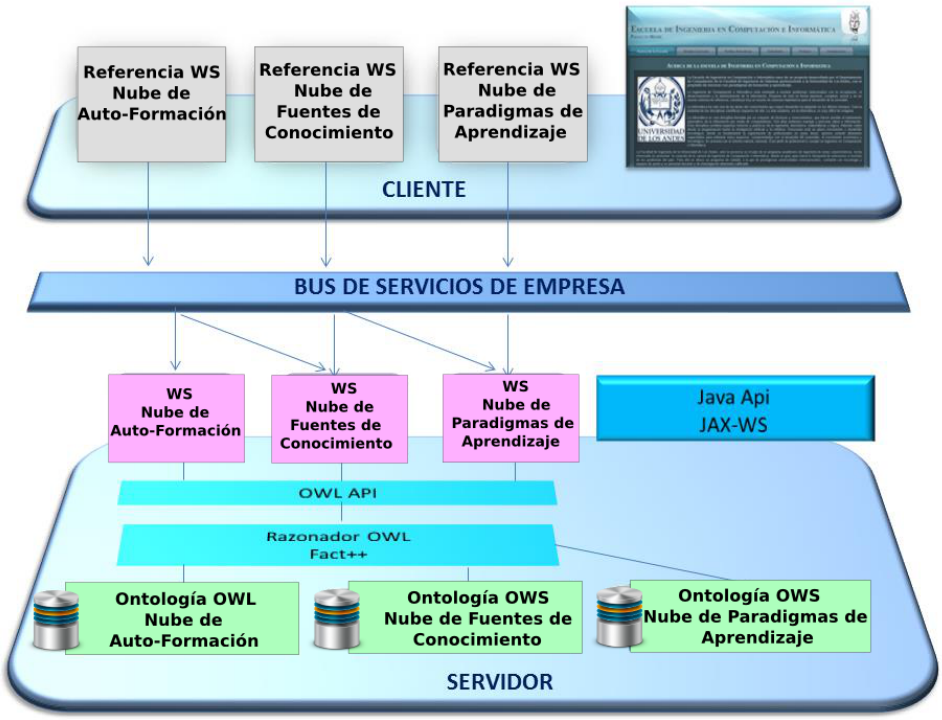

Figura 4.1: Arquitectura de la Plataforma Web

4.2. Fase de Desarrollo

4.2.1. Realización o Modelo Específico de la Plataforma (PSM)

Este modelo esta constituido por el diagrama de componente, para mostrar las divisiones y dependencias que tiene el sistema; el diagrama de paquetes, para describir la estructura lógica del sistema; y culmina con la ontología de desarrollo, que muestra las distintas tecnologías utilizadas y la interacción entre ellas.

4.2.1.1. Diagrama de Componente

La figura 4.2 muestra el diagrama de componente de la plataforma, conformado por tres grandes grupos de nodos. El primer grupo, lo integran los nodos que consumen los recursos ofrecidos por la nube de fuentes de conocimientos y esta compuesta por:

Interfaz Especialista: responsable de registrar los contenidos digitales y probar cada funcionalidad disponible en la nube de conocimiento compuesta por contenidos digitales. Este nodo posee dos paquetes importantes: el ClienteContenidoDigital.war, provee el servicio web con las paginas (coursera.jsp y generica.jsp) disponibles para el especialista, y las paginas para las pruebas de integración del sistema; El otro paquete, es el ClienteContenidoDigitalJUnitTest.java con las pruebas unitarias automatizadas del sistema.

Nube de Auto-Formación: solicita contenidos digitales específicos para el estilo de aprendizaje que tiene un estudiante particular en un tema determinado.

Módulo Adaptativo para el Estilo de Aprendizaje (en desarrollo): solicita las herramientas, instrumentos y actividades de un contenido digital, para verificar y actualizar el estilo de aprendizaje asociado a un estudiante.

El segundo grupo representa las funcionalidades de la nube de fuentes de conocimiento para los contenidos digitales, constituidos por los dos nodos siguientes:

FContenidoDigital: actúa como capa de seguridad, limitando los métodos disponibles en ContenidoDigital a solo los indicados en este servicio, funcionando como fachada entre el servicio de contenidos digitales y los clientes.

ContenidoDigital: es el núcleo de la plataforma de contenidos digitales, está conformado por el ContenidoDigital.java, que contiene la estructura de los métodos disponibles en el servicio; OntologiaContenidoDigital.java, encargado de todo el manejo de la ontología especifica del contenido digital; owlapi-distribution.jar, librería con capacidades para el manejo de la ontología; jfact.jar, razonador encargado de manejar las consultas semánticas para la ontología; y json-simple.jar, librería que permite la codificación y decodificación de textos de tipo JSON (formato ligero para el intercambio de datos).

Y el último grupo, lo componen tres nodos muy importantes para el funcionamiento de la nube de fuentes de conocimiento, ya que estos nodos contienen y entregan toda la información necesaria sobre los contenidos digitales. Cada nodo es detallado a continuación:

ContenidoDigital.owl: ontología con la información y estructura de los contenidos digitales certificados por el especialista.

Repositorios (Internet): son contenidos digitales buscados en la internet para ser ofrecidos por la nube de conocimientos como contenidos no certificados.

Nube de Paradigma de Aprendizajes: servicio que permite generar un conjunto de listas de herramientas, instrumentos y actividades asociados a un estilo de aprendizaje.

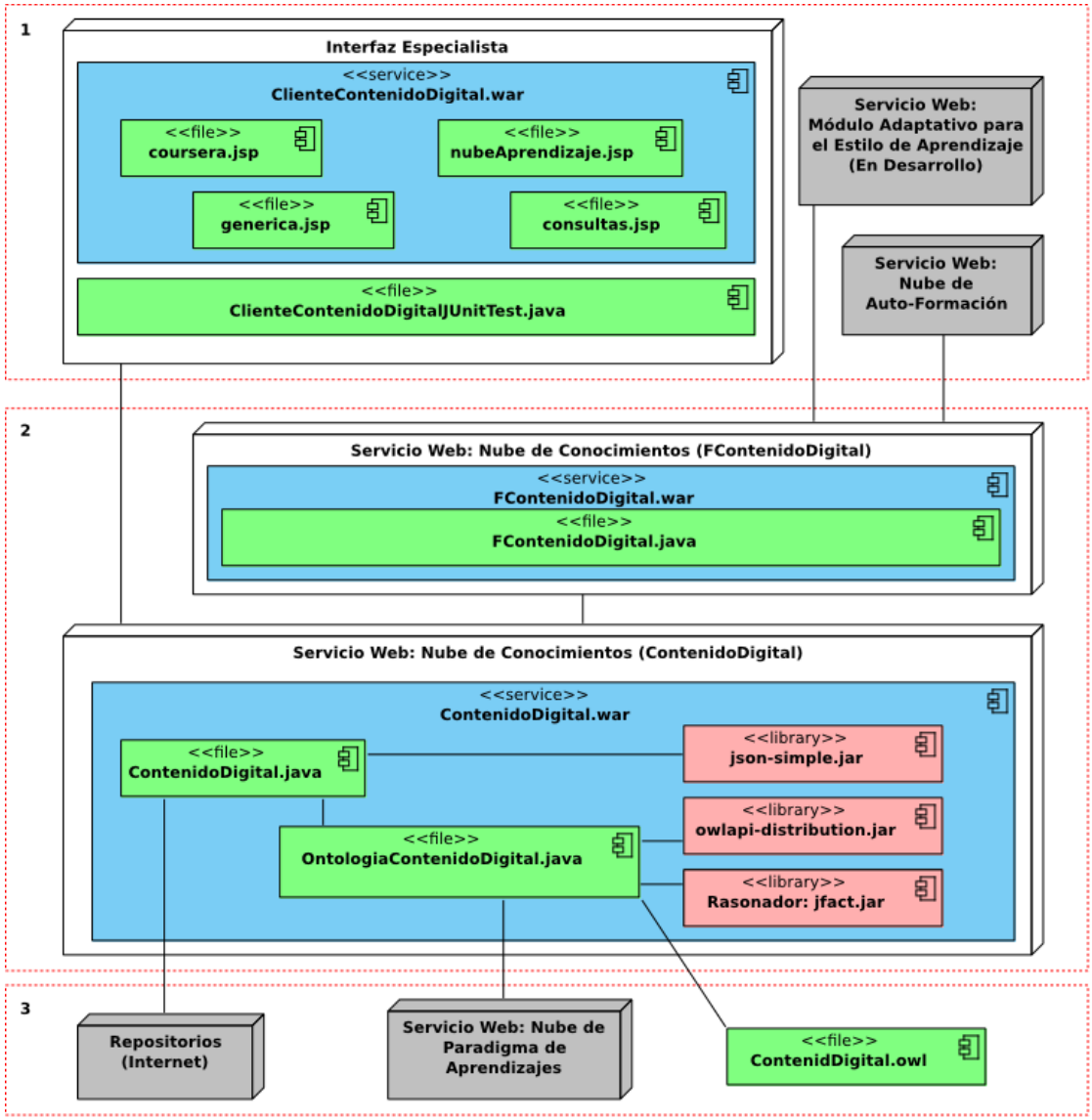

Figura 4.2: Diagrama de Componentes de la plataforma

4.2.1.2. Diagrama de Paquetes

Las figuras 4.3 y 4.4 muestran la descomposición lógica del sistema, organizados por paquetes que guardan coherencia con la estructura implementada en cada servicio. Los diagramas están compuestos por los siguientes paquetes:

Referencia Servicios Web: agrupa todos los servicios que son consumidos por la aplicación web. Para el caso del servidor, consume la NubeAprendizajeWS para obtener las herramientas, instrumentos y actividades para un estilo de aprendizaje, y para el cliente, consume ContenidosDigitales o FContenidoDigitales, dependiendo del nivel de acceso requerido.

Servicios Web: contiene todos los servicios creados por la aplicación web. El servidor implementa los servicios ContenidoDigital y FContenidoDigital, y el cliente no implementa ningún servicio.

Bibliotecas: librerías usadas para funcionar la aplicación web. El servidor usa jfact.jar, owlapi-distribution.jar y json-simple, y el cliente solo usa json-simple.jar.

Paquetes Fuentes: contiene todos los archivos fuentes de la aplicación web. Esta presente tanto en el servidor como en el cliente.

Archivos de Configuración: incluye los archivos necesarios para configurar adecuadamente la aplicación web. Esta presente tanto en el servidor como en el cliente.

Paginas Web: agrupa todas las paginas que funcionarán en el navegador web del cliente. Solo esta presente en el cliente.

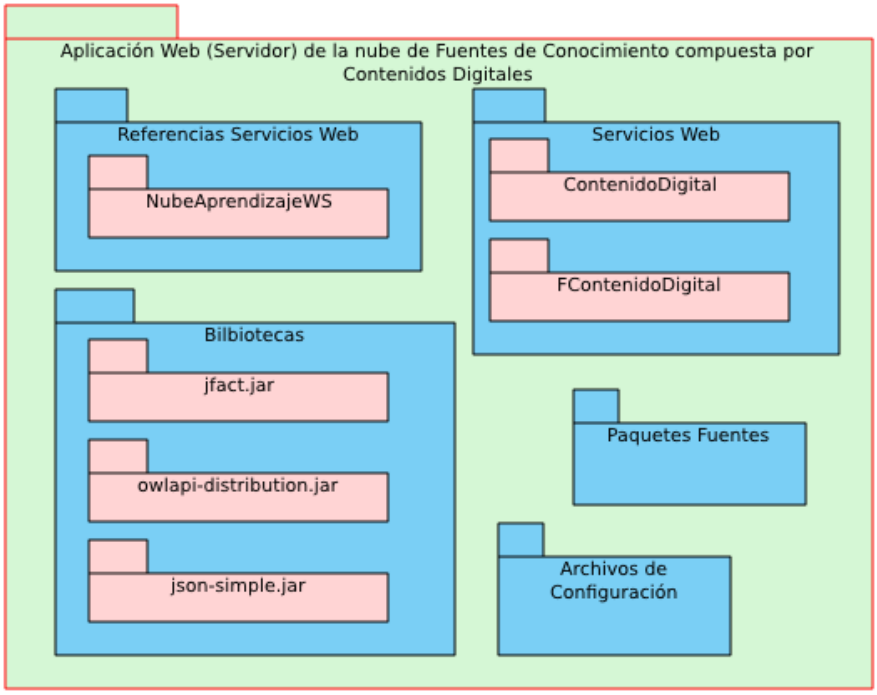

Figura 4.3: Diagrama de Paquetes de la plataforma (Servidor)

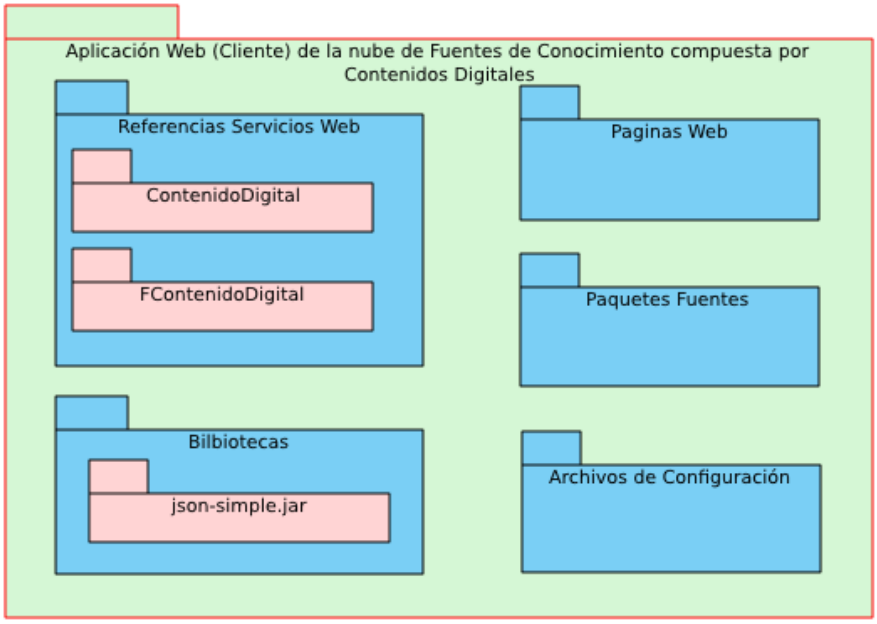

Figura 4.4: Diagrama de Paquetes de la plataforma (Cliente)

4.2.1.3. Ontología de Desarrollo

En la figura 4.5 se observan las distintas tecnologías utilizadas para construir y hacer funcionar la plataforma encargada de prestar los servicios relacionados con la nube de conocimientos compuestas de contenidos digitales. A continuación se muestra el glosario de términos y la taxonomía de conceptos:

1. Construcción del glosario de términos

Glosario de términos de la ontología PSM		
Nombre	**Descripción**	**Tipo**
Cliente	Interfaces con capacidad de consumir los contenidos digitales	Concepto
Servidor	Sistemas con capacidad de ofrecer servicios para el manejo de los contenidos digitales	Concepto
Servicios Web	Tecnología que utiliza un conjunto de protocolos y estándares que sirven para intercambiar datos entre aplicaciones.	Concepto
Ontología	Esquema conceptual con capacidad de razonamiento e inferencia	Concepto
JAX-WS	API de Java para la creación de servicios web	Concepto
Apache Tomcat	Contenedor web con soporte de servlets y JSPs.	Concepto
Protégé	Editor de ontologías y framework para la construcción de sistemas inteligentes	Concepto
OWL-API	API de Java que permite crear y manipular ontologías OWL	Concepto
JFact	Razonador con capacidad de manejar consultas semánticas para las ontología OWL	Concepto

Tabla 4.1: Glosario de términos de la ontología PSM

2. Construcción de la taxonomía de conceptos

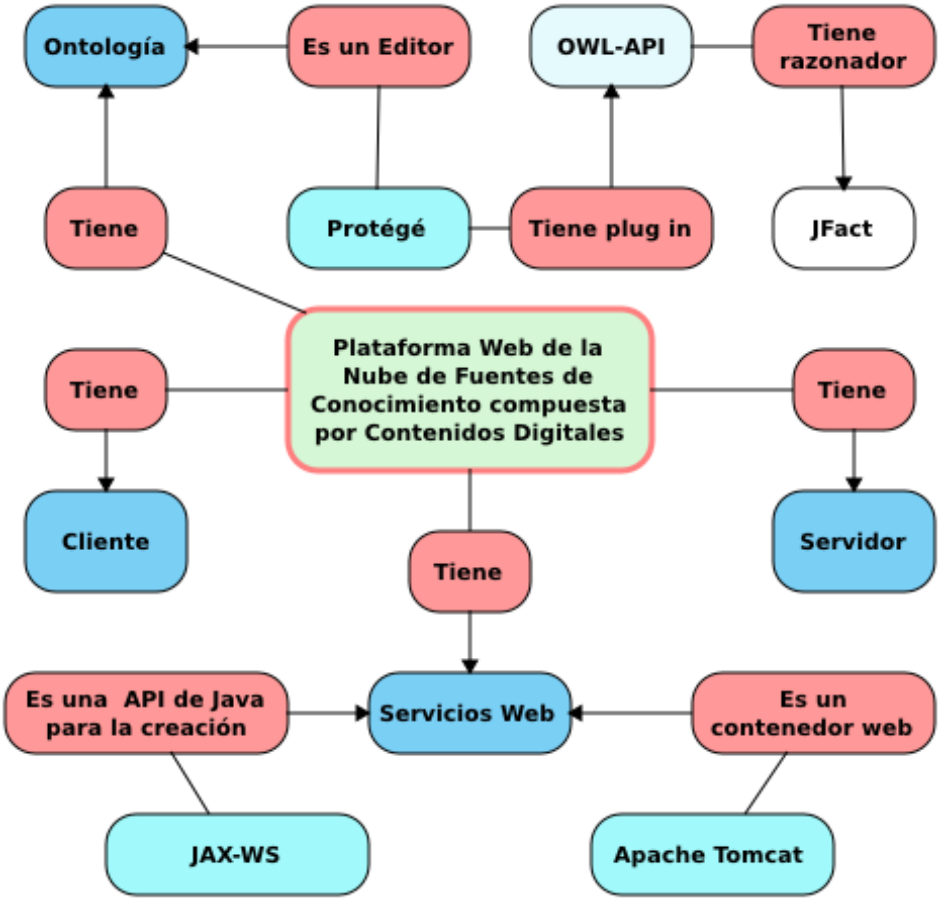

Figura 4.5: Ontología de Desarrollo de la Capa PSM de la plataforma

Capítulo V

Funcionamiento y Pruebas del Sistema

En este capítulo se procede a mostrar el prototipo y la evaluación del mismo. En la primera parte se muestran las capacidades del sistema desde dos manera de uso, y la segunda parte exhibe las distintas pruebas llevadas a cabo en el sistema para garantizar su correcto funcionamiento.

5.1. Funcionamiento del sistema

El funcionamiento del prototipo es mostrado desde la plataforma del proyecto madre y la interfaz para el especialista. Primero se muestra el prototipo integrado con el proyecto madre, que permite ofrecer contenidos digitales a la medida de las necesidades del estudiante, basado en su estilo de aprendizaje, y luego se muestra el funcionamiento de la interfaz para el especialista, la cual permite agregar información de los contenidos digitales certificados, y consultar los contenidos certificados y no certificados.

5.1.1. El Proyecto Madre

El sistema de contenidos digitales, a través de la nube de fuentes de conocimiento, se integra al proyecto madre como un nuevo servicio que aporta recursos a la nube de auto-formación. En la figura 5.1 se observa la interfaz de la nube de auto-formación, donde un estudiante consulta los módulos inscritos y tiene la opción "Ver Cursos en Línea", para solicitar contenidos digitales relacionados tanto con su estilo de aprendizaje como con el módulo inscrito.

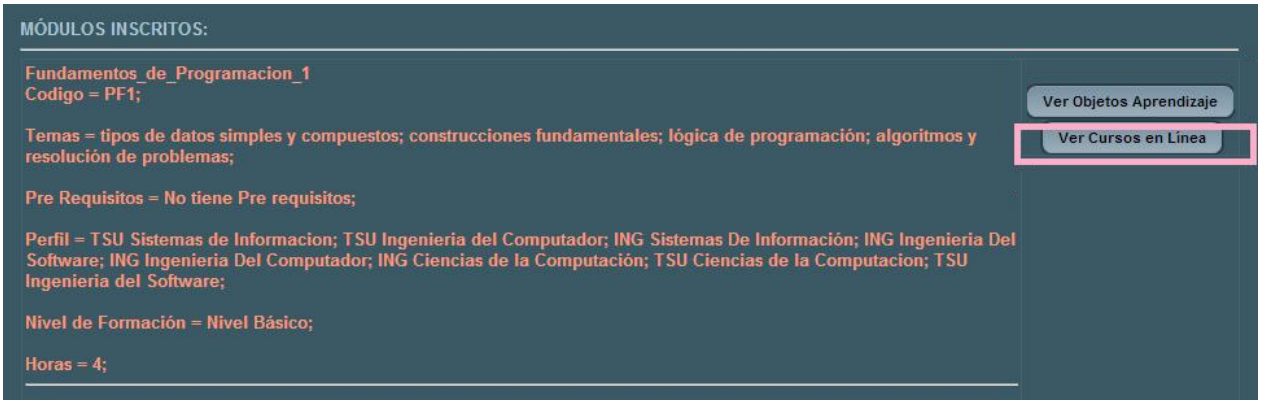

Figura 5.1: Interfaz de la Nube de Auto-Formación

Una vez elegida la opción "Ver Cursos en Línea", el sistema realiza una consulta de contenidos al servicio web de la nube de fuentes de conocimiento, pasando como parámetros el estilo de aprendizaje y módulo. El servicio responde a la solicitud con una lista de los contenidos más adecuados a la necesidad del estudiante según su estilo de aprendizaje, como se observa en la figura 5.2.

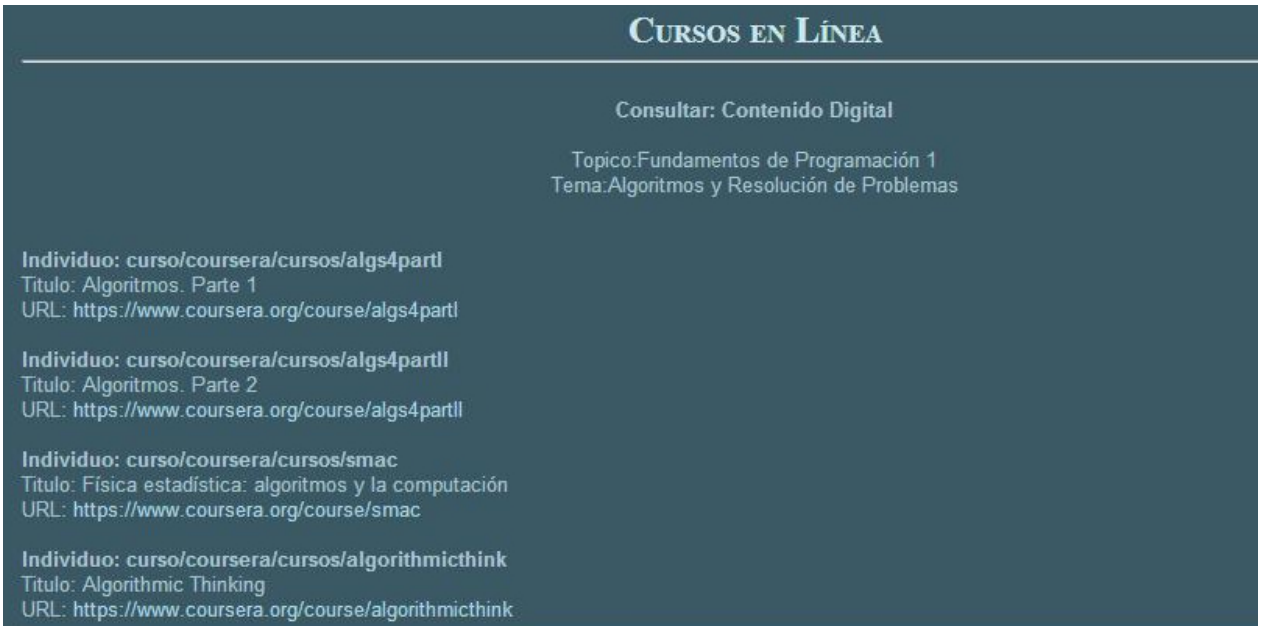

Figura 5.2: Ejemplo de Contenidos Recuperados por el Servicio de la Nube de Fuentes de Conocimiento desarrollado en este trabajo.

5.1.2. Interfaz Para el Especialista

El objetivo de esta interfaz es la de permiten agregar información sobre los contenidos digitales a la base de datos certificada que alberga dicha información. El sistema cuenta con tres módulos para el especialista, el primer modulo es una interfaz automatizada para la carga de contenidos, donde cada campo es llenado por la API de coursera, y complementado con información suministrada por el especialista. El segundo módulo es una interfaz manual para la carga de contenidos, donde cada campo debe ser llenado por el especialista, permitiendo agregar cualquier contenido disponible en la Web. Por último, cuenta con un módulo para consultas de contenidos certificados y no certificados.

5.1.2.1. Interfaz Automatizada

La interfaz automatizada permite minimizar los errores y aumentar la sencillez a la hora de agregar un contenido digital al sistema, y esto se logra gracias a la API que comparte Coursera para el desarrollo de aplicaciones que necesiten información provenientes de dicha fuente de consulta.

En la figura 5.3 se observa la interfaz, que esta compuesta por tres partes fundamentales. La primera, es su buscador de contenidos en coursera, que se presenta en la parte superior izquierda. La segunda, está debajo del buscador, que es donde se listan los contenidos encontrados. La última parte está a la derecha, que es donde muestran los datos del contenido digital que se está procesando, para finalmente ser guardado.

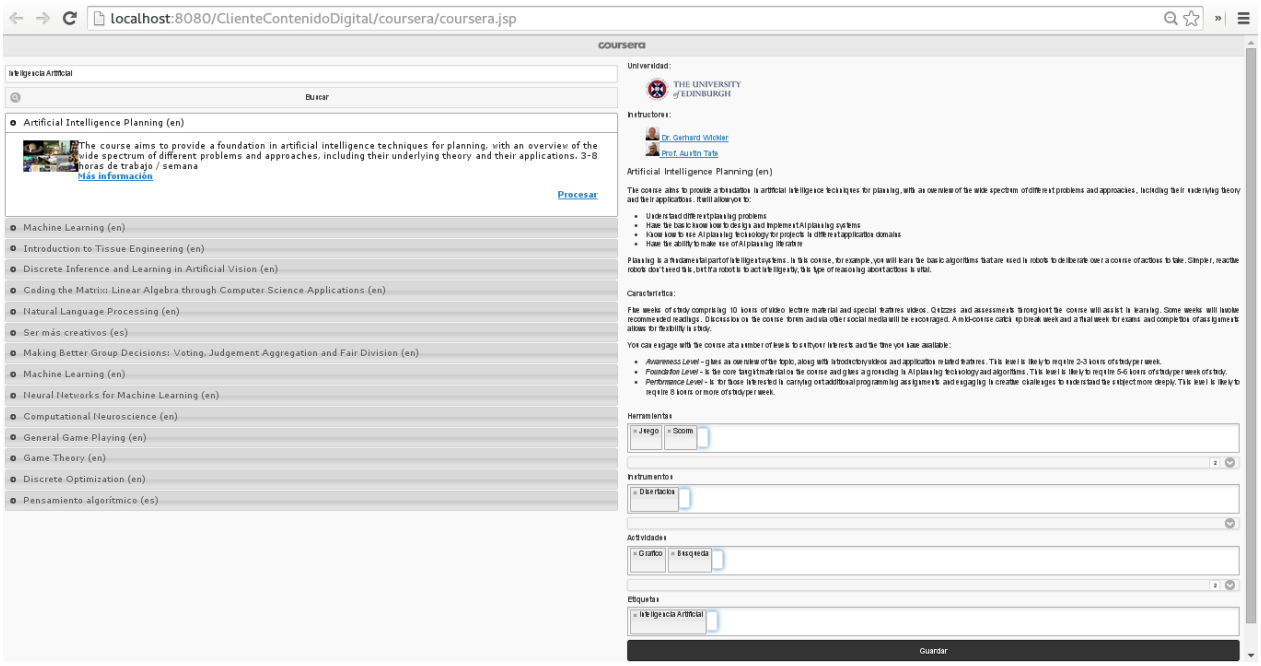

Figura 5.3: Interfaz Automatizada

A continuación se presenta el proceso que se sigue para guardar un contenido digital (ver figura 5.3):

1. Realizar la búsqueda deseada.
2. Seleccionar un contenido dentro de la lista mostrada, presionando el botón **Procesar.**
3. Revisar información encontrada sobre el contenido a procesar, mostrada en la parte derecha.
4. Llenar los campos: herramientas, instrumentos, actividades y etiquetas, muy importante, para luego encontrar contenidos personalizados según el estilo de aprendizaje.
5. Por último, almacenar toda la información, oprimiendo el botón **Guardar**.

5.1.2.2. Interfaz Genérica o Manual

La interfaz manual (ver figura 5.4) esta diseñada manteniendo el mismo diseño que la interfaz automatizada, con la diferencia que a la izquierda están todos los campos para colocar la información del contenido digital, y a la derecha muestra instrucciones de uso y

la información guardada en el sistema. Cada campo cuenta con su respectiva validación a la hora de ser llenados manualmente, un ejemplo de la validaciones que realiza la interfaz es la siguiente: si el especialista decide agregar un autor al contenido digital, este supervisa que los campos no vayan a estar vacíos a la hora de ser guardados, e informa de manera textual y visual dicho problema, cumpliendo con las características que se especifica en los requerimientos no funcionales como usabilidad, confiabilidad y seguridad del sistema.

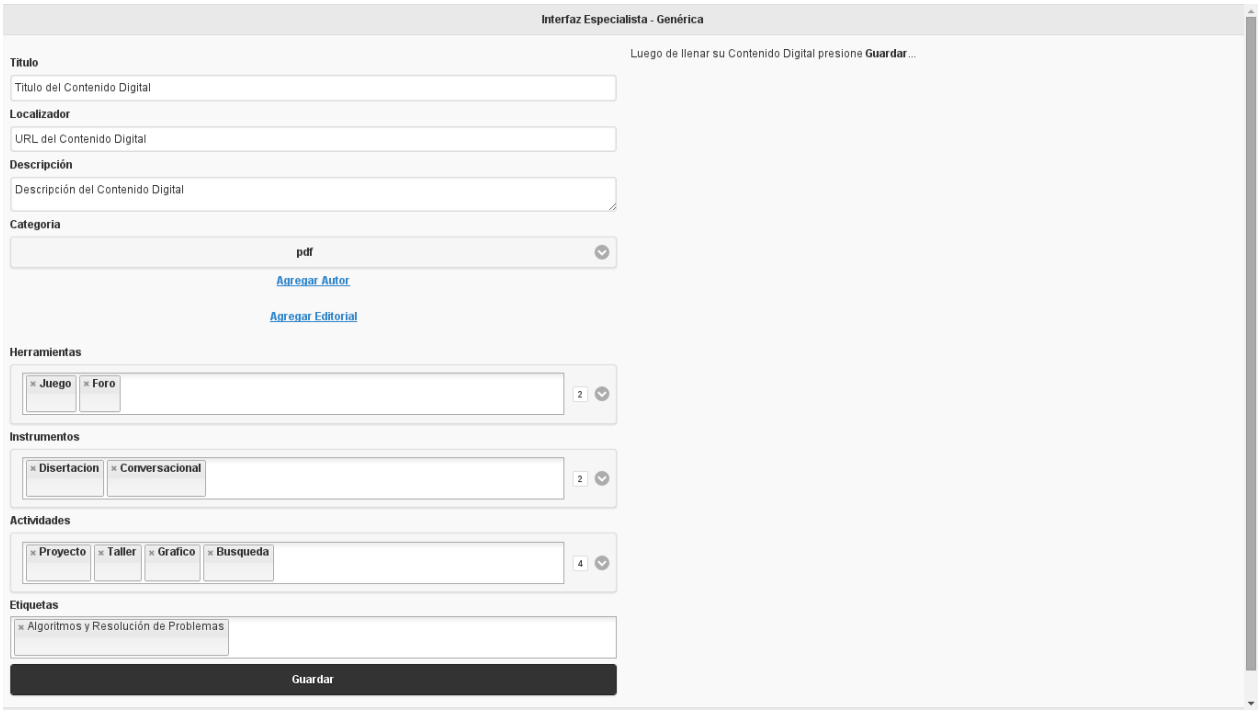

Figura 5.4: Interfaz Manual

5.1.2.3. Interfaz de Consultas

Esta interfaz permite verificar el correcto funcionamiento de las búsquedas de los contenidos digitales que estén tanto almacenados en la ontología, como los que están en la Internet.

La interfaz mostrada en la figura 5.5 está divida en dos columnas, la columna izquierda realiza consultas internas, es decir, busca contenidos digitales que se encuentra guardados en la ontología, y a la derecha consulta contenidos digitales en la Internet, gracias a los servicios ofrecidos por los buscadores externos.

Figura 5.5: Interfaz de Consultas

Ambas consultas necesitan tres parámetros para permitir realizar la búsqueda, estos parámetros son: el estilo de aprendizaje del estudiante, el tópico o asignatura que cursa el estudiante y el tema especifico dentro de la asignatura. Una vez lleno los campos, que pueden ser solo los de un lado, o ambos, se presiona su respectivo botón de consulta. La interfaz retorna el resultado en la parte inferior del botón, como una lista de contenidos digitales con su información asociada, por ejemplo: titulo, descripción, entre otras.

5.2. Pruebas de Sistema

La razón principal de hacer pruebas es garantizar que el sistema esté funcionando según lo solicitado. En esta sección se realizarán dos tipos de pruebas, una denominada unitarias y la otra de integración.

5.2.1. Pruebas Unitarias

Esta prueba permite comprobar que cada uno de los módulos funcione correctamente por separado. La idea es escribir casos de pruebas para cada método que integra el servicio web. Otro detalle que se quiere tener en cuenta, es que estas pruebas se puedan correr cada vez que se realiza un cambio en el sistema, para asegurar que conserva su misma funcionalidad, llamadas **pruebas de regresión**.

Para lograr estos objetivos, se procedió a implementar los casos de pruebas con ayuda del conjunto de bibliotecas de JUnit (ver figura 5.6), la esencia de esta biblioteca es realizar la

ejecución del Servicio Web de manera controlada, para poder evaluar si el funcionamiento de cada uno de los métodos del Servicio Web se comporta como se espera. Es decir, en función de algún valor de entrada se evalúa el valor de retorno esperado; si el Servicio Web cumple con la especificación, entonces JUnit devolverá que el método pasó exitosamente la prueba; en caso de que el valor esperado sea diferente al que regresó el método durante la ejecución, JUnit devolverá un fallo en el método correspondiente.

```
@Test
public void test0rden1_tituloAdd() {
     assertEquals( true , servicioCD.tituloAdd(curso,"titulo " + cadenaTest1) );
     assertEquals( false , servicioCD.tituloAdd(curso,"titulo " + cadenaTest2) );
     test0rden2_titulo();
}

@Test
public void test0rden2_titulo() {
     String titulo= servicioCD.titulo(curso);
     assertEquals( "titulo " + cadenaTest1, titulo );
}
```

Figura 5.6: Parte del código para la prueba automatizada con JUnit

Para el Servicio Web que gestiona los contenidos digitales se desarrollaron 31 casos de pruebas, para asegurar que cada método del servicio retornara los valores esperados en cada caso. Las pruebas se pueden identificar por el nombre, ya que después de la frase "testOrden_" indica que se esta probando. Por ejemplo, en la figura 5.6 se observan las siguientes pruebas: **testOrden1_tituloAdd** agrega el titulo de un contenido en la ontología y corrobora si tuvo éxito o no, y **testOrden2_titulo** consulta si el titulo de un contenido previamente almacenado esta correcto.

En la figura 5.7 se puede observar la salida que se obtiene de la ejecución de todas las pruebas automatizadas, con los tiempos de ejecución asociado a cada caso de prueba. Lo importante al pasar las pruebas, es que se garantizan que los contenidos se agregan (pruebas con terminación Add) a la ontología, y se verifica a través de los métodos de consulta, que la información sea la misma que fue agrega.

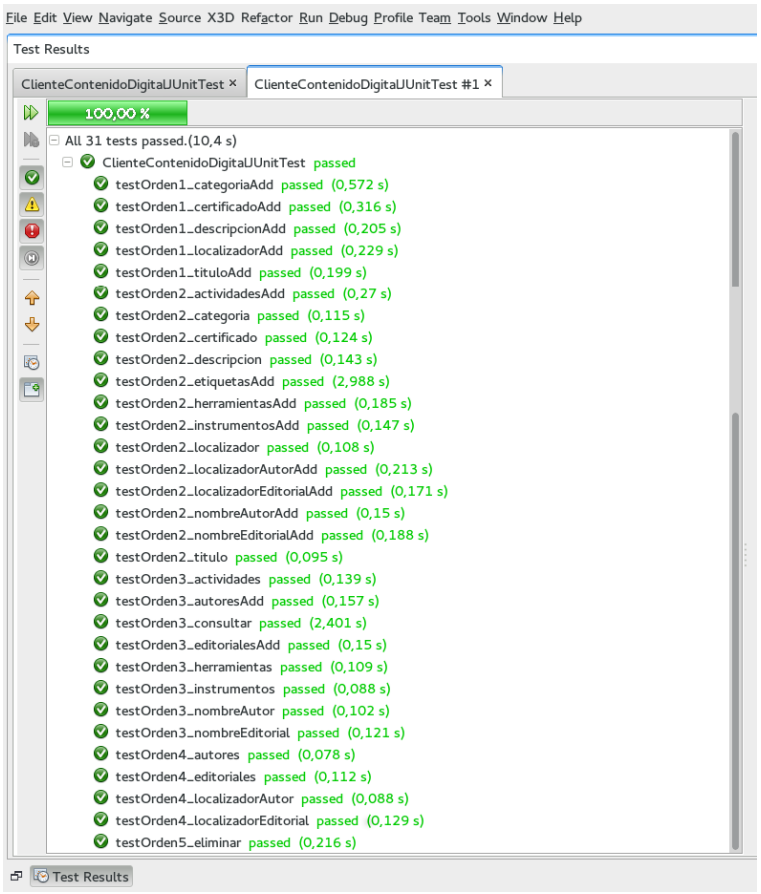

Figura 5.7: Resultado de las pruebas automatizadas con JUnit

5.2.2. Pruebas de Integración

El objetivo de las pruebas de integración, o pruebas integrales, es verificar el correcto ensamblaje entre los distintos componentes, una vez que han sido probados unitariamente, con el fin de comprobar que interactúan correctamente a través de sus interfaces, tanto internas como externas. Las interfaces internas son el conjunto de componentes que necesita o usa el sistema para funcionar adecuadamente, por ejemplo, el sistema usa al servicio que permite obtener las características de un estilo de aprendizaje. Las interfaces externas son el conjunto de componentes que utilizan al sistema, por ejemplo, la nube de auto-formación usa el servicio web de contenido digitales.

5.2.2.1. Interfaces Internas

La primera prueba es comprobar la interacción entre el servicio web que provee contenidos digitales y la nube de paradigmas de aprendizaje, la cual se realiza consultando al método nubeAprendizaje disponible en el servicio web de los contenidos digitales. El método solo recibe como parámetro el estilo de aprendizaje, como se observa en la figura 5.8, y este método invoca a los cuatro métodos que provee el servicio de paradigma de aprendizajes, que son los siguientes: verParadigma, verActividades, verHerramientas y verInstrumentos. El resultado que se obtiene de la prueba es toda la información relacionada con un estilo de aprendizajes.

```
List<String> nubeAprendizaje(int estiloAprendizaje) throws Exception {
      List<String> listaAprendizaje = new ArrayList<String>();

      NubeAprendizajeWS_Service service = new NubeAprendizajeWS_Service();
      NubeAprendizajeWS port = service.getNubeAprendizajeWSPort();

      listaAprendizaje.add(port.verParadigma(estiloAprendizaje));
      listaAprendizaje.add(port.verActividades(estiloAprendizaje));
      listaAprendizaje.add(port.verHerramientas(estiloAprendizaje));
      listaAprendizaje.add(port.verInstrumentos(estiloAprendizaje));

      return listaAprendizaje;
  }
```

Figura 5.8: Método nubeAprendizaje del servicio web del sistema

La figura 5.9 muestra el resultado de la consulta al método nubeAprendizaje, pasando como parámetro el estilo de aprendizaje 4534, que corresponde a un paradigma Activo. Como se lee en la figura, este retornar el paradigma Activo con sus herramientas, actividades e instrumentos, quedando adicionalmente verificada la comunicación entre los componentes del sistema buscador de contenidos y la nube de paradigmas de aprendizaje.

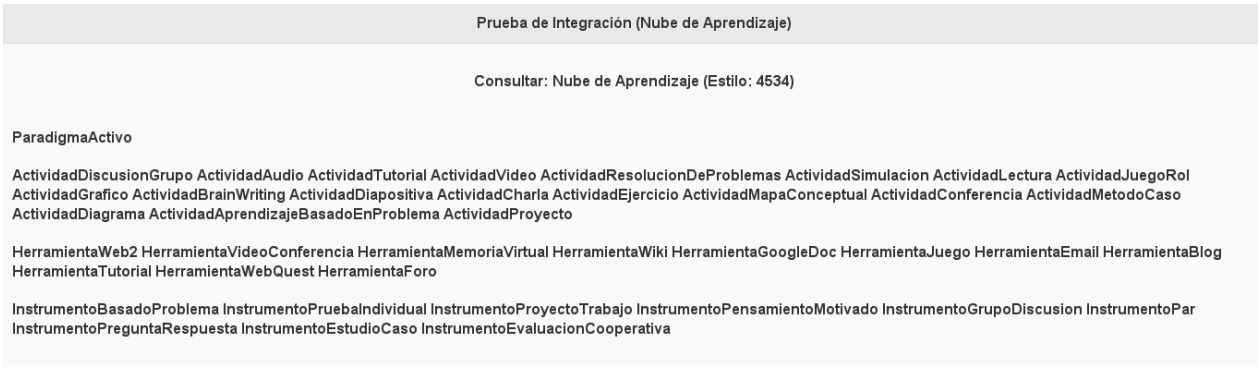

Figura 5.9: Prueba de Integración con la Nube de Aprendizaje

La siguiente prueba busca verificar la integración de la capa de seguridad que provee el servicio web FContenidoDigital al servicio web ContenidoDigital de la nube de fuentes de conocimiento. El proceso que se sigue es invocar los métodos disponibles en FContenidoDigital, que solo contiene métodos de consulta, y esta a su vez, invocan los métodos del servicio web ContenidoDigital correspondientes. El resultado que se debe obtener es poder consultar los contenidos digitales de la nube de fuentes de conocimiento. En la figura 5.10 se observa como se consume el servicio web FContenidoDigital, y la invocación del método consultar con los parámetros de pruebas: estilo, tópico y tema.

```
fcontenidodigital.FContenidoDigital_Service service = new fcontenidodigital.FContenidoDigital_Service();
fcontenidodigital.FContenidoDigital port = service.getFContenidoDigitalPort();

out.println("<center><br><b>Consultar: Contenido Digital</b></center>");

int estilo=4534;
topico="Fundamentos de Programación 1";
tema="Algoritmos y Resolución de Problemas";

List<String> listaContenidosDigitales = port.consultar(estilo,topico, tema);
```

Figura 5.10: Invocación del servicio web FContenidoDigital

El resultado final de toda la invocación de los métodos se muestran en la figura 5.11, donde se detalla la información como: Titulo, URL, Descripción, Herramientas, etc; asociado a uno de los contenidos digitales encontrados con el método consultar.

Figura 5.11: Resultado de la prueba al servicio FContenidoDigital

Por último, la prueba interna más importante es la comunicación desde la interfaz del especialista al servicio de contenidos digitales, en esta prueba se verifica la capacidad de búsqueda del sistema de la nube de fuentes de contenidos digitales, tanto para consultas internas como consultas externas. Ambas consultas necesitan tres parámetros para realizar la búsqueda, estos parámetros son: estilo, tópico y tema. La interfaz también posee validaciones (ver figura 5.12) que permiten minimizar los errores a la hora de realizar dichas consultas, donde todos los parámetros son obligatorios y el estilo debe ser un número de cuatro dígitos sin espacio.

Figura 5.12: Validaciones de los campos para las consultas

Los valores de pruebas para el caso de la **consulta interna** son los mismos que se han estado usando en anteriores pruebas, como se puede observar en la figura 5.13. La consulta externa se usa como tópico "Inteligencia Artificial" y tema "Planificación". El resultado que se debe obtener para cada caso es un listado de contenidos digitales con su información asociada.

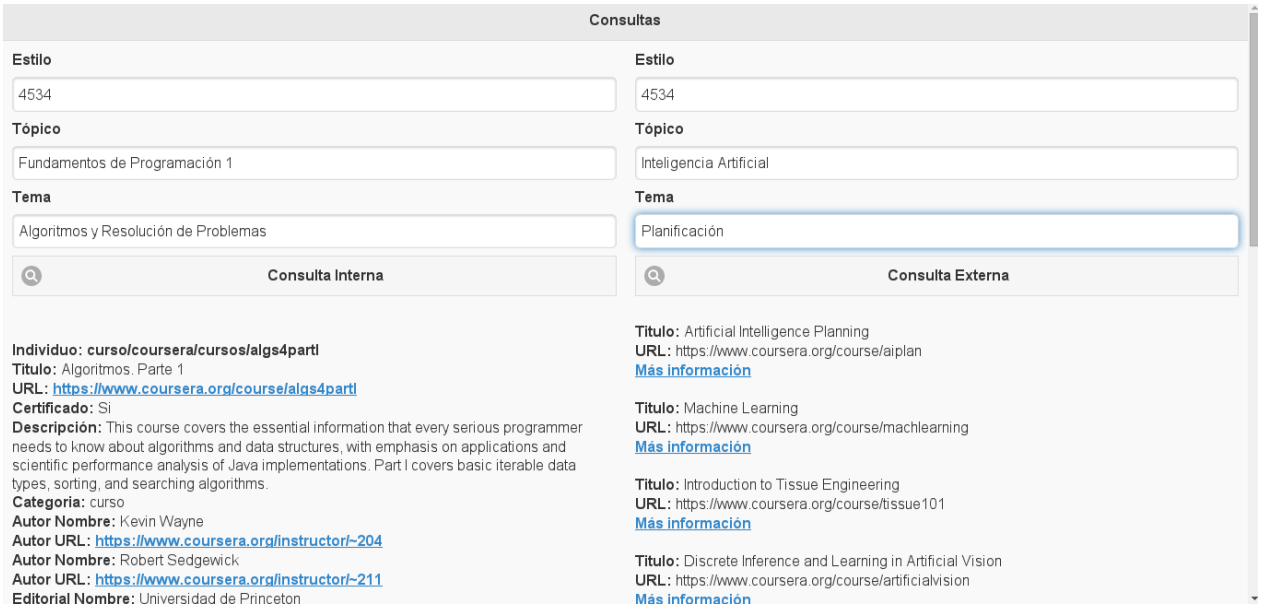

Figura 5.13: Resultados de las consultas

En la figura 5.13 también se puede apreciar el resultado de cada consulta, la cual es mostrado en la parte inferior de las misma. El grupo de contenidos digitales resultado de la

consulta interna se estudiará en la prueba de integración con el proyecto madre en la sección Interfaces Externas. En el caso de la **consulta externa**, la tabla 5.1 muestra un resumen de los contenidos encontrados, utilizando buscadores externos como el que ofrece Coursera a través de su API. En dicha tabla se observa que los contenidos guardan relación con los términos (resaltado con negrilla y cursiva) buscados, hay que resaltar que la calidad de esta búsqueda depende del buscador externo que ofrece la fuente de contenidos coursera y por dicho motivo son considerados como contenidos digitales no certificados.

Contenido	Resumen
Artificial Intelligence Planning (aiplan)	El curso tiene como objetivo proporcionar una base en técnicas de ***inteligencia artificial*** para la ***planificación***, con una visión general de la amplia gama de problemas y enfoques, incluyendo teoría subyacente y sus aplicaciones. CATEGORÍAS: Ciencias de la Computación - ***Inteligencia Artificial***
Machine Learning (Machlearning)	En este curso se estudia el aprendizaje automático, también conocida como la minería de datos, reconocimiento de patrones y análisis predictivo. Los Algoritmos de aprendizaje automático puede encontrar la manera de realizar tareas importantes, por generalizar, a partir de ejemplos. El aprendizaje automático, también conocida como la minería de datos, reconocimiento de patrones y análisis predictivo. CATEGORÍAS: Ciencias de la Computación - ***Inteligencia Artificial***, Análisis de Datos y Estadística
Introduction to Tissue Engineering (Tissue101)	Este curso estudia el campo de la ingeniería de tejidos, el cual esta en rápida expansión en la biología aplicada y la ingeniería biomédica, en ella se pretende crear órganos ***artificiales*** para el trasplante, la investigación básica, y/o el desarrollo de fármacos. CATEGORÍAS: Biología y Ciencias Naturales, Ingeniería
Discrete Inference and Learning in ***Artificial*** Vision (Artificialvision)	Este curso estudia aplicaciones de la visión ***artificial***, como la detección de objetos en imágenes naturales y la segmentación automática de muestras médicas, se basan en modelos que interpretan la información visual proporcionada a un ordenador. CATEGORÍAS: Ciencias de la Computación - ***Inteligencia Artificial***

Tabla 5.1: Contenidos digitales no certificados y sus características.

5.2.2.2. Interfaces Externas

El objetivo de esta prueba es comprobar que la nube de auto-formación consume la información contenida en el servicio web de los contenidos digitales a través del servicio FContenidoDigital, que brinda la seguridad de limitar los métodos a solo los necesarios para este nivel de cliente. Para realizar esta prueba se usa un estudiante con estilo de aprendizaje 4534, que corresponde al paradigma activo, la figura 5.9 detalla las características de este paradigma. Otro dato importante a considera en la prueba es el tópico y tema a consultar desde la nube de auto-formación, el tópico a usar es "Fundamentos de Programación 1", y el tema es "Algoritmos y Resolución de Problemas". La respuesta que se debe obtener es un listado de contenidos digitales adecuados a estas necesidades.

En la figura 5.14 se observa el resultado de la consulta desde la nube de auto-formación a los contenidos digitales. El proceso que sigue la consulta, se presenta a continuación:

1. Solicitar contenidos desde la nube de auto-formación del proyecto madre
2. Recibir la solicitud por parte del servicio web FContenidoDigital y hacer un llamado al servicio web ContenidoDigital
3. Buscar los contenidos según el tema y tópico, en el servicio web ContenidoDigital
4. Solicitar a la nube de aprendizaje las herramientas, actividades e instrumentos, pasando como parámetro el estilo de aprendizaje
5. Organizar y clasificar los contenidos digitales
6. Retornar toda la información al estudiante

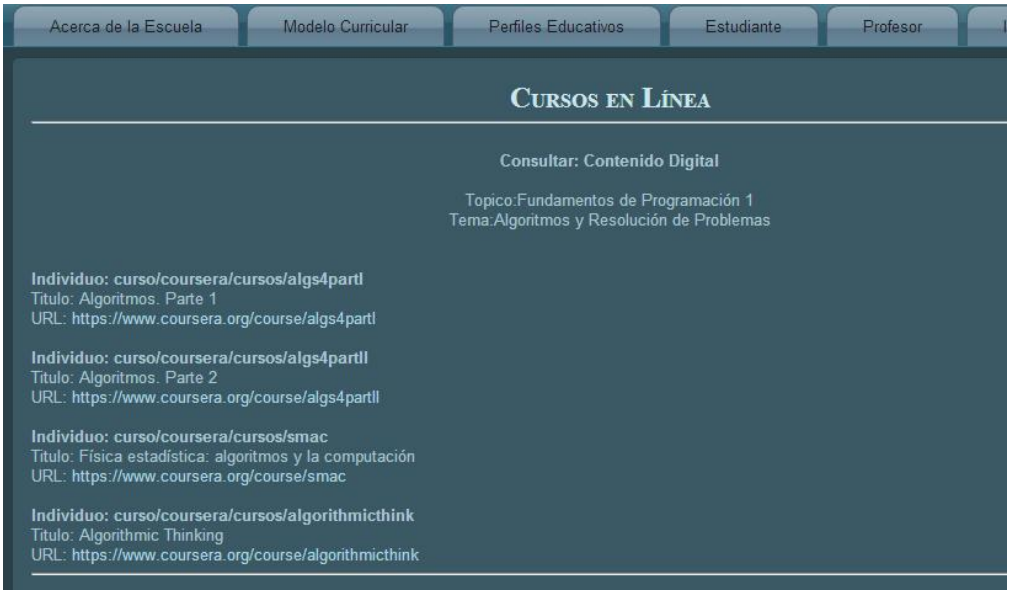

Figura 5.14: Interfaz de la Nube de Auto-Formación

Para determinar si la respuesta de la prueba es correcta, solo faltaría verificar que los contenidos cumplan con las características solicitadas, en la tabla 5.2 se detallan los contenidos digitales en el mismo orden en el que fueron encontrados por el sistema.

Contenido	Resumen	Características
algs4partI	En este curso aprenderás varios principios fundamentales del diseño de algoritmos : métodos de divide y vencerás, algoritmos de grafos, estructuras de datos prácticos (montones, tablas hash, árboles de búsqueda), algoritmos aleatorios, y por supuesto algo más.	ACTIVIDAD: ***Conferencia***, Ejercicio, Autoevaluación, ***Vídeo***, Examen y ***Lectura***. HERRAMIENTA: ***VideoConferencia***, ***Foro*** y ***Web2***. INSTRUMENTO: ***PruebaIndividual***
algs4partII	En este curso aprenderás varios principios fundamentales de diseño avanzado algoritmo: algoritmos y aplicaciones codiciosos; programación y aplicaciones dinámicas; NP-completo y lo que significa para el diseñador de algoritmo; el diseño y análisis de heurística; y más.	ACTIVIDAD: ***Conferencia***, Ejercicio, Autoevaluación, ***Vídeo***, Examen y ***Lectura***. HERRAMIENTA: ***VideoConferencia***, ***Foro*** y ***Web2***. INSTRUMENTO: ***PruebaIndividual***.
Sigue...		

Tabla 5.2: Contenidos digitales y sus características. (continua...)

Contenido	Resumen	Características
smac	Este curso analiza el enfoque computacional en la física moderna de una manera clara y accesible. En gran medida se basa en algoritmos concisos (cada uno entre 10 y 50 líneas de código Python).	ACTIVIDAD: Ejercicio, Autoevaluación, ***Vídeo***, Examen y ***Lectura***. HERRAMIENTA: ***Foro*** y ***Web2***. INSTRUMENTO: ***PruebaIndividual*** y ExamenTradicional
algortihmicthink	Este curso está diseñado para formar a los alumnos en los conceptos matemáticos y proceso de "pensamiento algorítmico", que les permite construir soluciones más simples y más eficientes para los problemas computacionales.	ACTIVIDAD: Ejercicio, ***Vídeo***, Examen y ***Lectura***. HERRAMIENTA: ***Web2*** INSTRUMENTO: ***EvaluacionCooperativa***.

Tabla 5.2: Contenidos digitales y sus características.

Una de las características que se debe cumplir, es que los contenidos coinciden con algunas de las propiedades del paradigma Activo (figura 5.15) como son: actividades, herramientas e instrumentos y que estos contenidos están ordenados de mayor a menor coincidencia. Para esta verificación se observa la tabla de contenidos, donde el primer contenido coincide con siete de estas propiedades (resaltado con negrilla y cursiva) y que van disminuyendo las coincidencias hasta llegar a la última con cuatro coincidencias. La otra característica a considerar, es que los contenidos deben estar relacionados con el tópico y tema pedido, para esto se verifica el resumen de cada contenido en la tabla, donde se puede apreciar que si guardan relación con lo buscado.

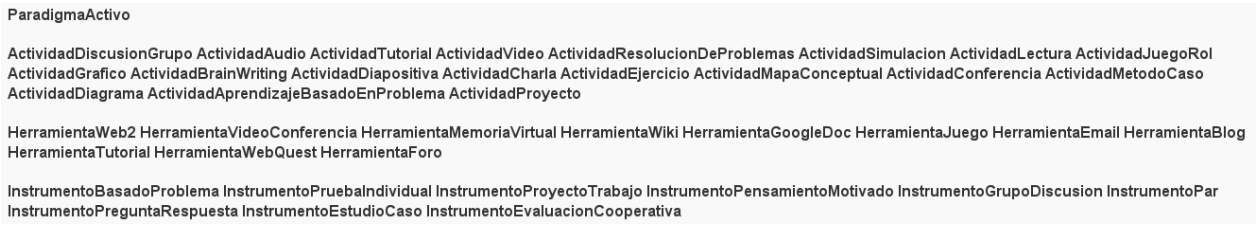

Figura 5.15: Paradigma Activo

5.3. Comparación con otros trabajos

La comparación de la calidad del sistema buscador de contenidos digitales, con respecto a otros sistemas que posean características similares, se muestran en la tabla 5.3 considerando

los siguientes criterios:

Filtros para la búsqueda: Capacidad del sistema para ajustar los resultados de los contenidos digitales ofrecidos.

Gestión de Información: Capacidad del sistema centrada en los procesos relacionados con el almacenamiento, el tratamiento y difusión del conocimiento explícito, que se encuentra representado en la metadata de los contenidos digitales.

Aprendizaje: Capacidad del sistema para aprender.

Interfaz: Capacidad del sistema para ofrecer distintas interfaces.

Tipos de archivos: Capacidad del sistema para clasificar los distintos contenidos digitales.

Interoperabilidad: Capacidad del sistema para comunicarse con otros sistemas.

Adaptabilidad: Capacidad del sistema para ajustarse a los distintos cambios del entorno.

Escalabilidad: Capacidad del sistema para crecer sin perder calidad en los servicios ofrecidos.

Los tres sistemas usados para la comparación, se basan en la gestión de los metadatos y no en el almacenaje de los contenidos digitales, ya que es la característica principal en el sistema buscador de contenidos desarrollado. El primero es Europeana, es la biblioteca digital europea, de acceso libre, que reúne contribuciones ya digitalizadas de reconocidas instituciones culturales de los 28 estados miembros de la Unión Europea. El segundo es BDU^2, es un proyecto iniciado por el SIU (Sistema de Información Universitario) de Argentina, que permite reunir recursos de información de valor académico de libre disponibilidad para el usuario final. El último es DigitalNZ Search, es una iniciativa con cerca de 200 socios, liderados por la Biblioteca Nacional de Nueva Zelanda, que buscan fomentar la reutilización de los contenidos digitales.

Sistemas / Criterios	Europeana	bdu[2]	DigitalNZ Search	Este trabajo
Filtros para la búsqueda	Tipo de archivo, idioma, fecha, país, copyright y proveedor	Tipo de archivo, autores, títulos y materias	Tipo de archivo, proveedor, fecha y copyright	Estilo de aprendizaje
Gestión de Información	Ontología y búsquedas semánticas	Ontología y búsquedas semánticas	Ontología y búsquedas semánticas	Ontología y búsquedas semánticas
Aprendizaje	No Posee	No Posee	No Posee	No implementado actualmente
Interfaz	Pagina Web, API, Widget	Pagina Web	Pagina Web, API	Pagina Web, Servicio Web
Tipos de archivos	Texto, imagen, vídeos, sonidos y 3d	Cualquier tipo de archivo alojado en los repositorios	Cualquier tipo de archivo alojado en los repositorios	Cualquier tipo de contenido alojado en la internet
Interoperabilidad	JavaScript Object Notation (JSON)	Open Archives Initiative Protocol for Metadata Harvesting (OAI-PMH)	Open Archives Initiative Protocol for Metadata Harvesting (OAI-PMH), XML Sitemap, API y RSS feed	JavaScript Object Notation (JSON)
Adaptabilidad	IND.	IND.	IND.	Permite agregar nuevos filtros de búsquedas y nuevas clasificaciones en los contenidos
Escalabilidad	Permite agregar nuevas fuentes de búsquedas	Permite agregar nuevos repositorios	Permite agregar nuevas fuentes de búsquedas	Permite agregar nuevas fuentes de búsquedas y nuevas características en los contenidos

Tabla 5.3: Tabla comparativa de sistemas desarrollados.

5.3 Comparación con otros trabajos

En cuanto al criterio del filtro de búsqueda, la diferencia con este trabajo, es porque usan filtros muy típicos como: tipo de archivo, titulo, etc. En cambio, el sistema buscador de contenido utiliza un filtrado basado en el estilo de aprendizaje, lo que permite entregar contenidos más adaptados a las capacidades de aprendizaje particulares de una persona.

En relación con el aprendizaje, ninguno de los sistemas poseen mecanismos que permita aprender, aunque la ontología de este trabajo ya posee conceptos que permite en una futura actualización incluirlo, por ejemplo, el concepto "Índice", que permitiría al cliente valorar los contenidos entregados en las búsquedas, permitiendo aumentar o disminuir su prioridad en próximas búsquedas de los clientes en el sistema.

En el resto de los criterios son muy semejante las características de los sistemas. En gestión de información, todos usan ontología para almacenar los metadatos, y realizar búsquedas semánticas. En interfaz, todos usan la pagina web como principal entorno gráfico, y se apoyan en otras interfaces para permitir la comunicación con otros sistemas. En tipos de archivos, como los sistemas manejan metadatos pueden utilizar cualquier tipo de contenido que puedan referenciar. En interoperabilidad, todos implementan mecanismos para compartir los metadatos de los contenidos digitales con otros sistemas. En adaptabilidad, no se logró determinar las capacidades adaptativas de los sistemas a comparar. En escalabilidad, todos permiten incluir nuevas fuentes de búsquedas y/o repositorios.

Capítulo VI

Conclusiones y Recomendaciones

6.1. Conclusiones

La presente investigación logró desarrollar el sistema buscador de contenidos digitales usando el paradigma ODA, para esto se especificaron las capas CIM, PIM y PSM, apoyados en los diversos diagramas UML, culminando en cada capa con su ontología especifica. El sistema se desarrolló como servicio web, permitiendo la interoperabilidad entre los distintos sistemas que conforman la plataforma del Proyecto Madre, y para el manejo de la ontología y procesos de razonamiento, se uso OWL-API y JFact, respectivamente. Por último, el estándar Ontology for Media Resource permitió manejar la información sobre las distintas características de los contenidos digitales, ampliando el estándar con nuevas características particulares, para ofrecer los contenidos según los distintos paradigma de aprendizajes.

En lo que se refiere a los contenidos digitales, se caracterizó la información que los describen, para ofrecer capacidades de búsquedas especificas según los distintos instrumentos, herramientas y actividades que tenga un estilo de aprendizaje. Esto se logró de manera general, al considerar las siguientes características: titulo, autor, localización, tipo de contenido, entre otros. De igual manera, se agrego detalles considerando características más especificas, para permitir identificar el contenido apropiado para un estilo de paradigma.

Con respecto a los estándares ontológicos, se evaluó el estándar de la W3C "Ontology for Media Resource 1.0", ya que permite registrar información de recursos almacenados en la Internet. Con dicho estándar se logró almacenar las distintas características de los conte-

nidos digitales, así mismo, se amplio el estándar para permitir inferencias sobre las nuevas características relacionadas con los estilos de aprendizajes

En relación con los servicios web, estos ofrecen innumerables beneficios para lograr la comunicación entre aplicaciones desarrolladas en diferentes lenguajes y en diferentes plataformas. Esto permitió obtener un sistema al cual se puede acceder a través de una única interfaz, logrando mejorar la flexibilidad para agregar servicios a la medida de las necesidades del Proyecto Madre. En el caso particular del sistema buscador de contenidos digitales, pasa a ser un servicio web añadido al Proyecto Madre, respondiendo a la necesidad de los contenidos digitales específicos a un estudiante con su estilo de aprendizaje particular.

Otra capacidad del sistema bastante interesante a destacar, es la manera en la que se implementó cada método o función del servicio web del sistema en relación con la información que se desea obtener. Cada método tiene asignado una sola responsabilidad, lo que permite a los clientes del sistema utilizar cada aspecto que necesita adquirir del servicio web de los contenidos, optimizando la cantidad de información intercambiada en cada solicitud al servicio web.

Este trabajo es innovador, ya que provee toda una plataforma para la búsqueda de contenidos más adecuados a cada estilo de aprendizaje que tenga un estudiante, entregando los contenidos priorizados según la mayor coincidencia con su estilo, logrando el apoyo en su proceso de auto-aprendizaje. También, el sistema permite su uso en otras plataforma educativas, gracias a que esta implementado con un servicio web, siempre y cuando se respete los parámetros de búsqueda del mismo.

6.2. Recomendaciones

A continuación se presentan un conjunto de posibles mejoras, trabajos futuros y recomendaciones:

- La interfaz automatizada que trabaja con la api de coursera, actualmente no recoge de manera automática la información sobre las herramientas, instrumentos y actividades

del contenido digital, se recomienda implementar un algoritmo que permita obtener dicha información a través de la descripción del formato del curso que provee coursera.

- Agregar la capacidad para cargar información de contenidos publicados en la Internet a través de distintos formatos de metadatos multimedia, como por ejemplo: OGG, LOM, entre otros. En la página de la ontología de la W3C, muestra el mapeo adecuado de cada campo desde metadatos multimedia hacia el estándar ontológico.

- La ontología cuenta con dos conceptos que no se han explotado, que podrían refinar aun más las búsquedas. La primera es "Audiencia", donde se podría especificar información del publico objetivo, y el otro concepto es "Índice", que permitiría al cliente valorar los contenidos entregados en las búsquedas, permitiendo aumentar o disminuir su prioridad en próximas búsquedas de los clientes en el sistema.

- Con esta investigación se tiene que un contenido digital contiene herramientas, instrumentos y actividades compatible con distintos estilos de paradigmas de aprendizaje, la cual es entregado al estudiante según la coincidencia con su estilo. Actualmente, la ontología cuenta con la capacidad de almacenar información de las partes de un contenido digital, donde esas partes pueden contener información relacionada a un paradigma de aprendizaje, esto permitiría ofrecer solo las partes de un contenido que más se adapten a un estilo de estudiante.

- El sistema esta diseñado para usar múltiples fuentes externas, aunque actualmente está trabajando con la información proveniente de la api de coursera, se recomienda ampliar las fuentes externas para permitir enriquecer las búsquedas de las consultas externas.

Apéndice A

Caso de usos

A.1. Registrar

Caso de Uso: Registrar	
Código	CU-02
Descripción	El sistema debe registrar en la ontología la información de los Contenidos Digitales, como título, localizador, herramientas, actividades, instrumentos, entre otros.
Actores	Especialista
Pre-Condición	Generar el identificador del Contenido Digital
Condición de Éxito	Registrar información en la ontología
Condición de Fracaso	Error al registrar la información en la ontología
Flujo Normal	1.-El especialista invoca al método correspondiente 2.-El método comprueba los parametros y recibe la solicitud 3.-Registra el contenido en la ontología
Flujo Alternativo	2.A.-Si los parametros no estan correctos retorna error

Tabla A.1: Caso de uso "Registrar"

A.2. Eliminar

Caso de Uso: Eliminar	
Código	CU-03
Sigue...	

Tabla A.2: Caso de uso "Eliminar" (continua...)

Caso de Uso: Eliminar (cont.)	
Descripción	El sistema debe eliminar en la ontología la información de los Contenidos Digitales, como título, localizador, herramientas, actividades, instrumentos, entre otros.
Actores	Especialista
Pre-Condición	Conocer el identificador del Contenido Digital
Condición de Éxito	Información eliminada en la ontología
Condición de Fracaso	No se elimino la información en la ontología
Flujo Normal	1.-El especialista invoca al método de eliminar 2.-El método comprueba los parametros y recibe la solicitud 3.-Elimina el contenido en la ontología
Flujo Alternativo	2.A.-Si los parametros no estan correctos retorna error

Tabla A.2: Caso de uso "Eliminar"

Referencias Bibliográficas

[1] Jairo Fuentes. Implementar un prototipo de un sistema de gestión de la nube de aprendizaje usando el paradigma oda. Tesis de maestría, Universidad de Los Andes, Mérida, Venezuela, 2014.

[2] Omar Portilla. Plataforma basada en soa usando el paradigma oda (ontology driven architecture), para la gestión de una nube de fuentes de conocimiento compuesta por objetos de aprendizaje. Tesis de maestría, Universidad de Los Andes, Mérida, Venezuela, 2014.

[3] Karla Moreno. Prototipo de plataforma web para la gestión de la nube de auto-formación basado en ontologías. Trabajo Especial de Grado. Universidad de Los Andes, Mérida, Venezuela, 2014.

[4] Jacques Delors. La educación o la utopía necesaria. *Informe a la UNESCO de la Comisión Internacional sobre la Educación. La educación encierra un tesoro*, pp. 13–36, 1996. *En línea:* http://www.unesco.org/education/pdf/DELORS_S.PDF.

[5] G Torres Vargas y Beatriz Juárez Santamaría. Los sistemas de organización del conocimiento y el manejo de contenidos digitales. *Biblioteca Universitaria: revista de la Dirección General de Bibliotecas de la UNAM*, 11(1):3–12, 2008. *En línea:* http://revistas.unam.mx/index.php/rbu/article/view/24650.

[6] P Castells, E Pulido, C Carranza, M Rico, F Perdrix, E Piqué, J Cal, R Benjamins, J Contreras, J Lorés, et al. Neptuno: tecnologías de la web semántica para una hemeroteca digital. *España: Ministerio de Ciencia y Tecnología*, 1, 2004. *En línea:* http://nets.ii.uam.es/neptuno/publications/neptuno-interaccion04.pdf.

[7] Rogério Aparecido Sá Ramalho y Mariângela Spotti Lopes Fujita. Aplicabilidad de ontologías en bibliotecas digitales. *Anales de Documentación, Universidad de Murcia Espinardo*, 14(1):1–19, 2011. *En línea:* http://www.redalyc.org/pdf/635/63517100007.pdf.

[8] Alianis Pérez Sosa y Yuniel Eliades Proenza Arias. Búsqueda sobre catálogos basada en ontologías. *Revista Digital Sociedad de la Información*, (27), 2011. *En línea:* http://www.sociedadelainformacion.com/27/busqueda.pdf.

[9] Fabiano Ferreira de Castro y Plácida Leopoldina Ventura Amorim da Costa SANTOS. Uso das tecnologias na representação descritiva: o padrão de descrição bibliográfica semântica marcont initiative nos ambientes informacionais digitais. *Ciência da Informação, Brasília*, 38(1):74–85, 2009. *En línea:* http://www.scielo.br/pdf/ci/v38n1/05.pdf.

[10] José Aguilar, Demián Gutiérrez, y Domingo Hernández. *Proyecto Madre*. Informe Técnico, Departamento de Ciencias Computacionales, Universidad de los Andes, Merida, 2012.

[11] Russell Shackelford, Andrew McGettrick, Robert Sloan, Heikki Topi, Gordon Davies, Reza Kamali, James Cross, John Impagliazzo, Richard LeBlanc, y Barry Lunt. Computing curricula 2005: The overview report. *ACM SIGCSE Bulletin*, 38(1):456–457, 2006. *En línea:* http://www.acm.org/education/education/curric_vols/CC2005-March06Final.pdf.

[12] Thomas Erl. *Service-oriented Architecture: A Field Guide to Integrating XML and Web Services*. Charles F. Goldfarb definitive XML series. Prentice Hall PTR, 2004.

[13] David Booth, Hugo Haas, Francis McCabe, Eric Newcomer, Michael Champion, Chris Ferris, y David Orchard. Web services architecture, w3c working group note, 11 february 2004. *W3C Consortium*, 2014. *En línea:* http://www.w3.org/TR/ws-arch/.

[14] Barry & Associates Inc. Web services explained. *Service Architecture*, 2014. *En línea:* http://www.service-architecture.com/articles/web-services/web_services_explained.html.

[15] Erik Christensen, Francisco Curbera, Greg Meredith, Sanjiva Weerawarana, et al. Web services description language (wsdl) 1.1, w3c note 15 march 2001. *W3C Consortium*, 2001. *En línea:* http://www.w3.org/TR/wsdl.

[16] John Colgrave, Karsten Januszewski, L Clément, y Tony Rogers. Using wsdl in a uddi registry, version 2.0. 2. *Technical note, OASIS*, 2004. *En línea:* https://www.oasis-open.org/committees/uddi-spec/doc/tn/uddi-spec-tc-tn-wsdl-v2.htm.

[17] Thomas R Gruber. A translation approach to portable ontology specifications. *Knowledge acquisition*, 5(2):199–220, 1993. *En línea:* http://tomgruber.org/writing/ontolingua-kaj-1993.htm.

[18] Tom Gruber. Ontology. *Entry in the encyclopedia of database systems*, 2008. *En línea:* http://tomgruber.org/writing/ontology-in-encyclopedia.htm.

[19] Jeff Heflin. Owl web ontology language-use cases and requirements. *W3C Recommendation*, 10:12, 2004. *En línea:* http://www.w3.org/TR/webont-req/.

[20] Riichiro Mizoguchi, Johan Vanwelkenhuysen, y Mitsuru Ikeda. Task ontology for reuse of problem solving knowledge. *Towards Very Large Knowledge Bases: Knowledge Building & Knowledge Sharing*, pp. 46–59, 1995.

[21] Gertjan Van Heijst, A Th Schreiber, y Bob J Wielinga. Using explicit ontologies in kbs development. *International journal of human-computer studies*, 46(2):183–292, 1997.

[22] Stuart Kent. Model driven engineering. En *Integrated formal methods*, pp. 286–298. Springer, 2002.

[23] OMG. Mda guide, version 1.0.1. *Object Management Group*, p. 62, 2003. *En línea:* http://www.omg.org/cgi-bin/doc?omg/03-06-01.pdf.

[24] Phil Tetlow, Jeff Z Pan, Daniel Oberle, Evan Wallace, Michael Uschold, y Elisa Kendall. Ontology driven architectures and potential uses of the semantic web in systems and software engineering. *W3C Working Draft*, 2005. *En línea:* http://www.w3.org/2001/sw/BestPractices/SE/ODA/.

[25] Jose Aguilar y Karla Moreno. Metodología para el desarrollo de aplicaciones basado en el paradigma oda. *CEMICID, Departamento de Computación, Universidad de Los Andes Mérida, Venezuela*, 2014.

[26] Ministerio de Tecnologías de la Información y las Comunicaciones. Primera cumbre de contenidos digitales. *Colombia 3.0*, 2011. *En línea:* http://www.mintic.gov.co/portal/604/w3-article-2334.html.

[27] Omar Nájera Ochoa. Tipología de contenido digital. *Educar*, 2004. *En línea:* http://portal.educ.ar/debates/educacionytic/inclusion-digital/tipologia-de-contenido-digital.php.

[28] Félix del Valle Gastaminza. Documentos digitales: hipertexto y edición digital. *Universidad Complutense de Madrid*, 2005. *En línea:* http://pendientedemigracion.ucm.es/info/multidoc/prof/fvalle/tema20.htm.

[29] W Lee, Tobias Bürger, Felix Sasaki, Véronique Malaisé, Florian Stegmaier, y J Söderberg. Ontology for media resource 1.0. *W3C working draft*, 2012. *En línea:* http://www.w3.org/TR/mediaont-10/.

[30] George Siemens. What is the theory that underpins our moocs? *Elearnspace*, 2012. *En línea:* http://www.elearnspace.org/blog/2012/06/03/what-is-the-theory-that-underpins-our-moocs/.

[31] We're empowering learning in the classroom and around the globe. *Edx inc*, 2015. *En línea:* https://www.edx.org/about-us.

[32] Forrest Wright. What do librarians need to know about moocs? *D-Lib Magazine*, 2013. *En línea:* http://www.dlib.org/dlib/march13/wright/03wright.print.html.

[33] *MirandaX*, 2015. *En línea:* https://www.miriadax.net.