



Universidad de las Ciencias Informáticas

Facultad 10

Trabajo de Diploma para optar por el título de Ingeniero en Ciencias Informáticas

Propuesta de un modelo de interoperabilidad entre los componentes de una Universidad Virtual para la Universidad de las Ciencias Informáticas

Autores:

Mairelis Méndez Osorio

Rolando Ernesto Sotomayor Alarcón

Tutores:

Msc. Daymy Tamayo Ávila

Ing. Roxana Cañizares González

Ing. Jesús Hidalgo Guillén

Ciudad de la Habana, 17 de junio del 2009.

DECLARACIÓN DE AUTORÍA

Declaramos ser autores de la presente tesis y reconocemos a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

Para que así conste firmamos la presente a los XX días del mes de Junio del año 2007.

Mairelis Méndez Osorio

Rolando Ernesto Sotomayor Alarcón

Firma de la Autora

Firma del Autor

Msc. Daymy Tamayo Ávila

Ing. Roxana Cañizares González

Ing. Jesús Hidalgo Guillén

Firma de la Tutora

Firma de la Tutora

Firma del Tutor

MSc. Daymy Tamayo Ávila: Especialista de la Dirección de Teleformación, Profesora de Metodología de Investigación. Facultad 10, Universidad de Ciencias Informáticas (UCI). Ciudad de La Habana. Cuba. daymy@uci.cu Ingeniera Informática, Universidad de Holguín, 2005. Máster en Informática Aplicada, UCI-CUJAE, 2007. Culminó sus estudios con índice académico de 4.94 puntos, obteniendo Título de Oro. Fue Alumna de Alto Aprovechamiento Académico y Alumna Ayudante en la Disciplina de Técnicas de Programación de Computadoras desde el segundo año de su carrera. Participó en un Proyecto de Educación a Distancia en la Universidad del 2001 hasta el 2005, año en el que comenzó a trabajar en la Universidad de Ciencias Informáticas. Ha participado en un gran número de eventos nacionales e internacionales. Ha sido tutora de varios Trabajos de Curso y Diploma, y de Alumnos Ayudantes. Actualmente perteneciente al Polo Teleformación, desempeñando el rol de arquitecta principal.

Ing. Roxana Cañizares González: Profesora de la asignatura de Práctica Profesional V. Facultad # 10, Universidad de las Ciencias Informática (UCI). Carretera a San Antonio de los Baños Km. 2 ½, Torrens, Boyeros, Ciudad de La Habana. Cuba. rcanizares@uci.cu Ingeniera en Ciencias Informáticas, UCI, 2007. Culminó sus estudios con un índice académico de 4.5 puntos. Fue Alumna Ayudante durante dos cursos en la Universidad de las Ciencias Informática en las Disciplinas de Ingeniería de Software, Metodología de la investigación y Seminario de Tesis. Se encontraba vinculada a proyectos productivos desde su tercer año de carrera, ocupando cargos de analista, jefa de proyecto y líder del grupo “Herramientas para la gestión de contenidos educativos reutilizables”, perteneciente al Polo Teleformación. Una vez graduada continuó en dicha universidad ocupando el cargo de Jefa del Polo Teleformación y vinculada a la docencia.

Ing. Jesús Hidalgo Guillén: Profesor de la asignatura de Introducción a la Programación. Facultad # 10, Universidad de las Ciencias Informática (UCI). Ciudad de La Habana. Cuba. jhidalgo@uci.cu Ingeniero en Ciencias Informáticas, UCI, 2007. Culminó sus estudios con un índice académico de 4.5 puntos. Fue Alumno Ayudante en la Universidad de las Ciencias Informática en las Disciplinas de Programación. Se encontraba vinculado a proyectos productivos desde su tercer año de carrera, ocupando cargos de desarrollador, jefe de proyecto y líder del grupo “Personalización de herramientas para la Teleformación”, perteneciente al Polo Teleformación. Una vez graduado continuó en dicha universidad ocupando el cargo de líder del grupo “Personalización de herramientas para la Teleformación” y vinculada a la docencia.

A mis padres por apoyarme siempre, por la dedicación y atención que han tenido conmigo durante toda mi vida y mis estudios.

A mi grupo 10503 por brindarme un espacio tan lindo donde estar.

A mis tutores por su dedicación y paciencia, en especial a Roxana Cañizares, por ayudarme tanto.

A todos los que de una forma u otra han aportado, al menos, con la simple pregunta ¿Cómo va la tesis?

Muchas Gracias.

Mairelis

A mis padres Orlando y Maria Isabel por ser una fuente inagotable de inspiración, por su apoyo, por su guía certera y por la educación que me han dado. Gracias además por su ejemplo día a día, su sacrificio, la confianza depositada en mí y su amor eterno. Les debo todo lo que soy. A mi hermano Orlandito, a quien adoro. Él también ha sido mi inspiración. A mi Abuela María por su apoyo incondicional, por considerarme y acogerme siempre como un hijo. A mi familia, quienes de una forma u otra me han ayudado a llegar hasta aquí. Gracias por su apoyo y confiar en mí.

A gley por su cariño, por ser mi amiga y comprenderme. A sus padres por todo su apoyo.

A mi duo de tesis (Mairelis), por su paciencia, entrega y dedicación en este trabajo. Gracias por tu esfuerzo y sacrificio.

A mis tutores por su apoyo, especialmente a Roxana.

A mis compañeros de proyecto.

A mi grupo de la universidad y todos los que han contribuido en mi formación profesional. en el transcurso de mis años de estudio.

A todos mis amigos, nunca olvidaré su ayuda en los momentos difíciles de mi carrera, son tantos que no quisiera pasar por alto el nombre de ninguno de ustedes, saben lo mucho que representan para mí. Gracias por compartir lo mejor de sí mismos y aceptarme en sus vidas.

A todos los que aún sin conocerme me brindaron desinteresadamente su ayuda.

A Fidel, a Raul, la UCI y la Revolución.

A todos, mi eterna gratitud.

Rolando

A mi mamá y mi papá porque de no ser por su amor, dedicación y sacrificio no hubiera sido posible hoy estar aquí, porque en cada hoja de este trabajo hay reflejada una palabra de aliento suya y me ayudaron a conseguir mi sueño y a pesar de estar lejos de mi, me cuidaron y mimaron como lo más preciado de su vida.

A mi novio Alexander que me ha apoyado muchísimo en estos cuatros años de relación.

A todas las personas que me brindaron una sonrisa o un gesto de cariño, y que contribuyeron con mi educación.

Gracias por existir y apoyarme en todo momento.

Mairelis.

A mis padres por su inmenso amor, su apoyo incondicional y por confiar en mí; por su sacrificio, entrega y por ser mi guía en este largo camino recorrido. A ellos porque se que éste era uno de sus sueños.

A mi hermano y a mi abuela.

A mi familia en general.

A gley por los momentos de felicidad y por permanecer a mi lado.

A mis amigos.

De Todos, es también este su resultado.

Rolando

La evolución del uso de las Tecnologías de la Información y las Comunicaciones (TICs) en la educación, ha potenciado un desarrollo exponencial del e-learning y con ello, la necesidad de una formación continua en niveles superiores de enseñanza, trasladando actividades de su entorno formal hacia la virtualidad, es así que surgen y se manifiestan como propuestas innovadoras, las Universidades Virtuales (UVs).

Innumerables herramientas de soporte al aprendizaje a distancia vía electrónica o digital han surgido con el objetivo de facilitar esta modalidad de enseñanza, como son: LMS, LCMS, Repositorios de Objetos de Aprendizaje (ROA), Herramientas de Autor (HA), Sistema de Gestión Académico (SGA), Bibliotecas Virtuales (BV), Laboratorios Virtuales (LV), etc. Para lograr un entorno integrado, donde todos estos componentes sean interoperables entre sí y mantengan una comunicación e intercambio de datos de forma segura se hace necesario el uso de estándares.

El presente trabajo propone un modelo para lograr la interoperabilidad entre componentes de una Universidad Virtual (UV) para la Universidad de las Ciencias Informáticas (UCI), teniendo en cuenta el envío seguro de los datos a través de los mismos. Con tales objetivos se hace un estudio en el que se identifican y proponen los componentes LMS, ROA, SGA, HA, BV, LV para la UV de la UCI, usando los estándares abiertos de interoperabilidad IMS-DRI, SQI, OKI mediante sus OSIDs, Z39.50, IMS Enterprise Services, IMS-LIP y WS-Security para tratar aquellos aspectos referentes a la seguridad en el sistema y se propone una arquitectura orientada a servicios que admite la inclusión y exclusión de otros componentes sin alterar el funcionamiento, flexibilidad y sencillez a la misma.

PALABRAS CLAVES

Componentes, Estándares, E-learning, Interoperabilidad, Modelo, Universidad Virtual.

INTRODUCCIÓN	1
CAPÍTULO 1. FUNDAMENTOS TEÓRICOS.	11
1. Introducción del Capítulo	11
1.2. ¿Qué es una Universidad Virtual?	11
1.2.1. Componentes de las Universidades Virtuales	13
1.2.1.1. Learning Management System (LMS)	13
1.2.1.2. Learning Content Management Systems (LCMS)	15
1.2.1.3. Herramientas de Autor	15
1.2.1.4. Repositorio de Objetos de Aprendizajes.....	16
1.2.1.5. Bibliotecas Digitales o Virtuales.....	17
1.2.1.6. Laboratorios Virtuales	19
1.2.1.7. Sistema de Gestión Académica (SGA).....	21
1.2.1.8. Secretaría Virtual	23
1.2.2. Ejemplos de Universidades Virtuales	24
1.2.2.1. Universidad Oberta de Cataluña (UOC)	24
1.2.2.2. UNED de España.....	26
1.2.2.3. UNED de Costa Rica.....	28
1.2.2.4. Universidad Virtual del Ministerio de Ciencia, Tecnología y Medio Ambiente (CITMATEL).....	29
1.3. ¿Qué es un estándar?	30
1.3.1. Estándares de interoperabilidad entre los componentes de las UV.....	31
1.3.1.1. Shareable Content Object Reference Model (SCORM).....	33
1.3.1.2. MACHine-Readable Cataloging (MARC 21)	35
1.3.1.3. Instructional Management System (IMS).....	36
1.3.1.3.1. IMS Digital Repositories Interoperability (IMS-DRI)	37
1.3.1.3.2. IMS Enterprise Services.....	39
1.3.1.3.3. IMS Learner Information Package (IMS LIP)	40
1.3.1.3.4. IMS Question & Test Interoperability	43
1.3.1.4. ANSI / NISO Z39.50.....	43
1.3.1.5. The Open Archives Initiative Protocol for Metadata Harvesting (OAI-PMH).....	45

1.3.1.6.	Simple Query Interface (SQI)	47
1.3.1.7.	Open Knowledge Initiative (OKI)	48
1.3.1.7.1.	Arquitectura Open Service Interface Definitions (OSID) de OKI	49
1.3.2.	Estándares de Seguridad	51
1.3.2.1.	Web Service Security (WS-Security, WSS)	52
1.3.2.2.	XML Encryption	53
1.3.2.3.	XML Digital Signature.....	54
1.3.2.4.	Extensible Rights Markup Language (XrML)	55
1.3.2.5.	Security Assertion Markup Language (SAML)	57
1.3.2.6.	XML Key Management (XKMS).....	58
1.3.2.7.	Extensible Access Control Markup Language (XACML)	59
1.4.	Arquitectura Orientada a Servicio	59
1.4.1.	Características de SOA	60
1.4.2.	Niveles de Madurez de SOA.....	63
1.4.3.	¿Qué son los Servicios Web?.....	64
1.4.4.	Estándares empleados por los servicios web	65
1.5.	Componentes de UVs existentes en la UCI	66
1.5.1.	Biblioteca Digital de la UCI	66
1.5.2.	Akademos.....	67
1.5.3.	Entorno Virtual de Aprendizaje (Moodle).....	70
1.5.4.	Repositorio de Objetos de Aprendizaje.....	71
1.5.5.	Herramientas de Autor Web (ROXS)	72
1.6.	Conclusiones del Capítulo	72
CAPÍTULO 2. DESCRIPCIÓN DE LA SOLUCIÓN PROPUESTA		74
2.	Introducción del Capítulo 2	74
2.1.	Modelo de la Universidad Virtual de la UCI (UVUCI).....	74
2.2.	Arquitectura de la UVUCI.....	78
2.3.	Descripción de Estándares utilizados en la UVUCI	82
2.3.1.	Descripción de los métodos de Simple Query Interface (SQI).....	82
2.3.2.	Las guías OSID de OKI.	84

2.3.2.1.	Métodos de la OSID Authentication	85
2.3.2.2.	Métodos de la OSID Authorization	86
2.3.2.3.	Métodos de la OSID Logging	90
2.3.3.	Descripción de los métodos de IMS Enterprise Services (IMS ES)	99
2.3.4.	Descripción del estándar IMS Learner Information Package (IMS LIP)	102
2.3.5.	Descripción de los métodos de IMS Digital Repositories Interoperability (IMS DRI)	115
2.3.6.	Descripción de los servicios ANSI / NISO Z39.50	116
2.3.7.	Estándar de Seguridad Web Service Security (WS-Security).....	126
2.4.	Niveles de madurez en la UVUCI.....	129
2.5.	Trabajo Futuro	130
2.6.	Conclusiones del Capítulo	131
CONCLUSIONES GENERALES.....		132
RECOMENDACIONES		133
BIBLIOGRAFÍAS.....		134
REFERENCIAS BIBLIOGRÁFICAS.....		134
BIBLIOGRAFÍA CONSULTADA.....		138
ANEXOS.....		143
GLOSARIO DE TÉRMINOS.....		191

<i>Figura 1 Un paquete de contenidos SCORM [29]</i>	35
<i>Figura 2. Arquitectura funcional. Reproducida de IMS DRI Information Model [31]</i>	39
<i>Figura 3. Estructura de datos básica de un paquete IMS LIP</i>	42
<i>Figura 4. Diagrama de bloques de un sistema Z39.50</i>	44
<i>Figura 5. Estructura de los módulos de Akademos [48]</i>	68
<i>Figura 6. Modelo de la UV para la UCI</i>	75
<i>Figura 7. Arquitectura de la UV para la UCI</i>	78
<i>Figura 8. Subcapa Servicio de Interoperabilidad de la Arquitectura de la UVUCI</i>	80
<i>Figura 9. Bus de Servicios</i>	81
<i>Figura 10. Estructura de un mensaje SOAP con WS Security</i>	126
<i>Figura 11. Ejemplo de un fragmento de código del estándar WS Security [56]</i>	128
<i>Figura 12. Nivel 2 de la UVUCI a través de una UDDI</i>	130

INTRODUCCIÓN

En nuestros días, las tecnologías de la información y las comunicaciones (TICs) favorecen el aprendizaje de estudiantes, técnicos, trabajadores, y profesionales en general. Los sistemas de educación a distancia, han dejado de ser sólo una alternativa más de enseñanza para convertirse en un modelo educativo de innovación en el presente siglo.

Es por ello que la sociedad de la información y el conocimiento demanda cada día y con mayor fuerza nuevas alternativas educacionales que resuelvan aquellos problemas referentes al progreso de la enseñanza, y que permita formas flexibles de aprendizaje con una participación activa individual y en grupos, acorde con la existencia global de un acceso desigual a las TICs.

Es precisamente en este entorno condicionado por las TICs, que los seres humanos interactúan con atractivas propuestas de formación tales como: el *e-learning*¹ (Electronic Learning).

Existen diferentes formas de definir e-learning, en el presente trabajo se tomará como el *“conjunto de tecnologías, aplicaciones y servicios orientados a facilitar la enseñanza y el aprendizaje a través de Internet/Intranet, que facilitan el acceso a la información y la comunicación con otros participantes”* [1]

Con el e-learning la institución educativa se desarrolla en un nuevo nivel, en el que la tecnología se convierte en una aliada para la formación profesional de los alumnos y el desarrollo continuo de los profesores. El poder de Internet por otro lado, pone a su servicio la posibilidad de crear redes educativas que anulan las limitaciones geográficas y temporales. Así mismo la combinación de la educación tradicional con las nuevas tecnologías de aprendizaje optimiza exponencialmente el proceso educativo [2].

Con la necesidad de una formación continua en niveles superiores de enseñanza en esta nueva sociedad de la información y aparejado al vertiginoso desarrollo de la informática y las redes de comunicación, muchas universidades comienzan a trasladar paulatinamente actividades de su entorno presencial hacia

¹ Traducido al español es Aprendizaje Electrónico pero para no variar su significado se citará en inglés.

la virtualidad². Se apuesta en estos espacios virtuales por una mejora y optimización continua de la gestión del conocimiento y los procesos de la enseñanza del aprendizaje.

Es precisamente, en este contexto, que surgen y se desarrollan aceleradamente nuevas formas de adquirir conocimientos basados en el e-learning, ejemplo de ello lo constituyen las Universidades Virtuales (UVs) o Campus Virtuales³ (CVs).

Existen varias formas de definir las UVs, en el presente trabajo se basa en *“una forma flexible de enseñanza especializada a distancia que utiliza avanzadas tecnologías de comunicaciones, esto permite ofrecer una formación de excelencia, cumpliendo con los estándares académicos internacionales”*[3]. Además genera nuevos espacios para que la comunidad universitaria pueda llevar a cabo sus procesos de comunicación, gestión, servicios e investigación.

Técnicamente, las UVs basan su funcionamiento en la integración de una serie de *componentes⁴ o piezas funcionales de un conjunto integrado*, así como el empleo de tecnologías como: estándares de interoperabilidad y seguridad, Web Services, Arquitectura Orientada a Servicio (SOA), etc. A pesar de ello, un número significativo de las UVs existentes, sólo hacen referencia a plataformas e-learning como los Learning Management Systems (LMS) o Sistemas para la Administración del Aprendizaje.

Estas plataformas permiten la creación y gestión de cursos sin la necesidad de conocimientos profundos de programación o de diseño gráfico, ya que brindan una interfaz cómoda para su confección y de gran accesibilidad para el usuario [4]. Independientemente de las ventajas y posibilidades que brinda este tipo de plataforma, las mismas carecen de herramientas⁵ que faciliten el aprovechamiento y gestión del aprendizaje optimizando así estos procesos. De tal modo esta desventaja pone en evidencia la imposibilidad de los LMS para establecerse como una UV.

² Proceso de representar electrónicamente y en forma numérica digital, los objetos que se encuentran en el mundo real.

³ En este documento se emplea el término Campus Virtuales para hacer referencia a una Universidad Virtual

⁴ Un componente se refiere a una pieza funcional de un conjunto integrado que son la UVs.

⁵ Repositorios de Objetos de Aprendizaje, Herramientas de Autor, Bibliotecas digitales y otras.

El contenido didáctico gestionado dentro las UVs se puede representar en pequeñas unidades que permiten ser archivadas, localizadas y reutilizadas en infinidad de contextos dentro de una UV. Dichas unidades reciben la denominación de Objetos de Aprendizaje (OA⁶).

Para una mejor gestión de los OA, estos son almacenados en un Repositorio de Objetos de Aprendizaje (ROA) los cuales “permiten realizar búsquedas eficientes a través de los metadatos” [5]. Para la creación de los OA se hace necesario el uso de las Herramientas de Autor (HA), las que permiten la creación de contenidos (es decir, OAs), asignarles características⁷ y la generación de los mismos a través de estándares, para lograr su reutilización. Estos dos componentes (ROA y HA) se pueden encontrar dentro de los Learning Content Management Systems (LCMS).

Los LCMS son plataformas que se presentan con regularidad dentro de una UV como componentes de las mismas, son sistemas que aprovechan las capacidades de creación y almacenamiento de contenidos de los CMS. A diferencia de los LMS, los LCMS permiten gestionar el desarrollo de contenidos, su acceso y almacenamiento posibilitando así que las UVs puedan prescindir de otros componentes como los LMS, ROA y HA dado que estas herramientas se encuentran contenidas dentro de un LCMS, lo que indudablemente permitiría conformar un sistema más sencillo en cuanto al número de componentes que se integran pero con menor riqueza en cuanto a funcionalidad sin mencionar además la lentitud de los LCMS propiciado por la complejidad en su implementación.

Las UVs brindan una extensa gama de funcionalidades mediante las cuales los usuarios del sistema pueden conocer sus evaluaciones, sus registros, horarios docentes, etc., a través de la utilización de los Sistemas de Gestión Académicas (SGA) los que proveen mecanismos automatizados que facilitan la planificación, organización, gestión y control académico.

Además, otro de los componentes principales de una UV es una Biblioteca Virtual (BV), que básicamente: *“es una biblioteca que tiene la tecnología que permite llevar millones de volúmenes enriquecidos con*

⁶ OA es “cualquier recurso con una intención formativa, compuesto de uno o varios elementos digitales, descrito con metadatos, que pueda ser utilizado y reutilizado dentro de un entorno e-learning”. [27].

⁷ Estas características se refieren a los metadatos.

sonido e imagen en movimiento y bases de datos a miles de kilómetros de distancia ocupando solo el espacio del computador” [6].

Otro componente que se añade en las UVs, permite controlar y gestionar las transacciones existentes entre miembros, afiliados y usuarios, con la plataforma, a la hora de realizar el proceso de matriculación a los cursos, consultar los expedientes académicos, calificaciones, estado de la solicitud de beca y el estado de la solicitud del título, una denominación común para dicho componente es el término de Secretaría Virtual (SV).

Para poner en práctica los conocimientos adquiridos en una UV es necesario contar con uno o varios Laboratorios Virtuales, en dependencia de las asignaturas que lo necesiten. Un Laboratorio Virtual (LV) *“es un sistema informático que pretende simular el ambiente de un laboratorio real y que mediante simulaciones interactivas permite desarrollar las prácticas de laboratorio que son de gran utilidad pues ayudan al usuario a desarrollar este tipo de habilidad (relacionada con el experimento) a distancia”*. [7]

Las UVs como se ha mencionado hasta el momento, disponen de un grupo considerable de componentes que deben establecerse como un todo para poder realizar el intercambio de información, este es el punto exacto en el que juega su papel la interoperabilidad, definida por la IEEE⁸ como *“la habilidad de dos o más sistemas o componentes para intercambiar información y para usar la información que ha sido intercambiada”*.

Para alcanzar la interoperabilidad se pueden utilizar un conjunto de estándares⁹ que son normas de obligatorio cumplimiento en un entorno determinado. Entre los estándares de interoperabilidad se destacan: Simple Query Interface (SQI), The Open Archives Initiative Protocol for Metadata Harvesting (OAI-PMH), Open Knowledge Initiative (OKI), Z39.50¹⁰, IMS Digital Repositories Interoperability (IMS-DRI) y otros. Además es recomendado el uso y consumo de los servicios web¹¹ que cada uno de estos

⁸ IEEE the Institute of Electrical and Electronics Engineers (<http://www.ieee.org>).

⁹ Es una norma, tipificación o un patrón de cómo realizar algo.

¹⁰ Z39.50 también conocida por “ANSI/NISO Z39.50”. Se explicará en el capítulo 1.

¹¹ Son una buena práctica de programación que permite intercambiar datos e información en una red de computadores.

componentes publique, siguiendo una Arquitectura Orientada a Servicio (SOA), que de acuerdo con [8], es una evolución de la llamada computación distribuida, basada en el paradigma de pregunta / respuesta para aplicaciones síncronas y asíncronas. En ellas las lógicas de negocio o las funciones individuales son presentadas como servicios para aplicaciones consumidoras / clientes.

Asimismo, son implementados dentro una UV estándares de seguridad como: Web Service Security (WS-Security, WSS), XML Encryption, XML Digital Signature, eXtensible Rights Markup Language (XrML), Security Assertion Markup Language (SAML), XML Key Management (XKMS) y otros, los cuales tienen por objetivo proveer mecanismos al sistema que garanticen una comunicación y transferencias de datos seguras.

Son disímiles las propuestas e iniciativas de UVs a nivel mundial. Destacan entre ellas de acuerdo a [9], por sus resultados, las excelentes críticas recibidas así como su posicionamiento en internet las universidades de: Universidad Oberta de Catalunya (UOC), el Instituto Tecnológico de Monterrey, la Universidad de Islas Baleares (UIB¹²), The Open University¹³ de Inglaterra, UNED¹⁴ en España y la UNED en Costa Rica.

En Cuba, varias son las instituciones que han invertido recurso y tiempo para contar con una UV, tal es el caso de la Universidad Virtual del Ministerio de Ciencia, Tecnología y Medio Ambiente (CITMA), como parte del Programa Ramal científico técnico del CITMA, nombrado Red de la Ciencia. Esta UV está integrada por un aula, una biblioteca, un museo, un laboratorio, algunos eventos, todos estos virtuales y gestionados a través de una oficina [4].

Otra institución que ha perseguido semejantes objetivos es el Ministerio de Salud (MINSAP), que ha accionado en este ámbito mediante la propuesta “Universidad Virtual de Salud (UVS)”, actualmente está soportada utilizando software libre. Sus principales componentes son: plataforma para la gestión docente centralizada, plataforma educativa, repositorio de contenidos que permite sindicación y los metadatos

¹² <http://www.uib.es/ca/>

¹³ <http://www.open.ac.uk/>

¹⁴ http://portal.uned.es/portal/page?_pageid=93,1,93_150635&_dad=portal&_schema=PORTAL

utilizan el estándar LOM¹⁵ para la referenciación y búsqueda [10]. Pero esta UV no es lo suficientemente interoperable pues los contenidos no son referenciados y no son diseñados como OA.

De gran interés resulta la Universidad de la Ciencias Informáticas (UCI), la que desde sus inicios se propone impulsar los procesos docentes en general (pregrado y postgrado), de la comunidad universitaria, de una manera diferente, flexible y para que no solamente fuesen empleados los métodos convencionales de enseñanza en los que intervienen la relación estudiante-profesor a través de un aula, haciendo uso de recursos físicos como elementos en formato duro y otros. Con tales propósitos queda resuelta a utilizar e implementar una serie de herramientas y aplicaciones digitales accesibles desde la red, aprovechando la infraestructura tecnológica disponible. Debe resaltarse que aunque la UCI no cuenta con una UV como tal, si posee gran parte de los recursos necesarios para matarizarla, de momento solo atesora atractivas iniciativas.

Una de las herramientas con mayor aceptación y resultados en cuanto a la gestión de los procesos de aprendizaje vía electrónica en la UCI, se da a conocer en el año 2005 con la implantación de un Entorno Virtual de Aprendizaje (EVA) basado en el LMS Modular Object-Oriented Dynamic Learning Environment (MOODLE¹⁶), una de las plataformas de aprendizajes más difundidos y exitosos a nivel mundial.

Paralelamente, un grupo de trabajo perteneciente al “Polo Teleformación”¹⁷ en la Facultad # 10 concibe y desarrolla las primeras versiones de las herramientas: Repositorio de Objetos de Aprendizaje (ROA, nombre representativo del repositorio) y Herramienta de Autor (ROXS) cuyos objetivos se centran en la gestión de OAs basados en el estándar SCORM 1.2¹⁸ y la creación de dichos OAs respectivamente.

¹⁵ Learning Object Meta-Data (LOM). Estándar utilizado para la descripción de los metadatos, con el objetivo de crear descripciones estructuradas de recursos educativos.

¹⁶ Entorno de Aprendizaje Dinámico Orientado a Objetos y Modular.

¹⁷ Grupo o área de Investigación + Desarrollo dividida en subgrupos cuyas labores se centran sobre líneas diferentes y/o Desarrollo, extensión, personalización, instalación y soporte de herramientas para el e-learning, basado en los estándares de mayor aceptación internacionalmente, teniendo en cuenta criterios arquitectónicos modernos y Seguridad.

¹⁸ Shareable Content Object Reference Model (Modelo de Referencia de Objetos de Contenido Compartibles) orientado a empaquetar OA descritos mediante LOM.

Ambas herramientas tienen como propósito la gestión de los recursos didácticos utilizados en el proceso docente, pero las mismas no implementaban una vía de comunicación directa entre ellas, teniendo como único enlace el intercambio de OAs, exponiendo así una gran limitante al no estar enfocadas en lograr la interoperabilidad.

A raíz de esta limitante surge una investigación en la que obtiene un Framework Integrador¹⁹ para la gestión de contenidos educativos reutilizables (FI), el mismo tiene en cuenta la incorporación de estándares como SQL e IMS-DRI pero no es hasta la concepción de éste, que tales estándares son incluidos a ROA en pos de lograr una comunicación con el LMS. Dicho FI está basado en estándares de interoperabilidad e igualmente facilita, que herramientas para la gestión de contenidos tales como: un LMS, un Repositorio de Objetos de Aprendizaje, y una Herramienta de Autor se comuniquen. Este FI propone que las herramientas ROA y ROXS sean desarrolladas nuevamente, teniendo en cuenta una arquitectura orientada a servicios y en el caso del LMS propiciar su integración a los dos componentes mencionados con anterioridad mediante pasarelas [4].

El FI a pesar de estar orientado a la interoperabilidad entre sistemas que gestionen contenidos educativos puede ser tomado como una primera aproximación a desarrollar e implementar una UV, pues el mismo sólo contempla la comunicación e interoperabilidad entre tres componentes básicos y necesarios en una UV (un LMS, un Repositorio de Objetos de Aprendizaje, y una Herramienta de Autor). Otros componentes como las SV, los LV, los SGA y las BV son excluidos pues no estaban dentro de los objetivos de este Framework. Otro inconveniente a tener en cuenta en el FI es que no se hace hincapié en aspectos referentes al intercambio de información de forma segura.

Otro componente desarrollado en la UCI es Akademos, un SGA cuyo objetivo principal es proveer a la universidad mecanismos automatizados que faciliten la planificación, organización, gestión y control académico, encargado de mostrarles a los estudiantes su expediente, sus evaluaciones hasta la fecha y realizar búsquedas para acceder a las evaluaciones de otros estudiantes. En la última versión disponible

¹⁹ Representa una arquitectura de software referida a un dominio, de modo que modele las relaciones entre las entidades de dicho ámbito. [4]

en la universidad, sólo es capaz de comunicarse con el EVA para comunicar aspectos referentes a la estructura de los grupos docentes, no así para la actualización de evaluaciones realizadas en la plataforma, no utilizan ningún estándar de interoperabilidad y no es desarrollado basado en una arquitectura orientada a servicio.

Por otra parte la biblioteca de tipo automática²⁰, existente en la UCI, permite proporcionar servicios especializados de búsqueda en catálogos y en bases de datos internacionales sobre disímiles temas de la especialidad y productos de información de alto valor agregado que apoyan las funciones principales de la universidad referentes a los procesos de: docencia, investigación, extensión universitaria y producción, mediante la selección adecuada de las fuentes de información y la utilización de las tecnologías de la información y las comunicaciones [11]. Esta biblioteca está en un proceso de cambio y de transformación en la que se plantean mejoras a través del uso de un conjunto de estándares que faciliten el intercambio y la interoperabilidad, por ejemplo el MARC 21²¹ que define un formato de datos para el resultado de consultas bibliográficas y Z39.50²² para la recuperación de información basada en la estructura cliente/servidor, pero en estas mejoras no se incluye una implementación basada en una arquitectura orientada a servicio.

De manera general todas las herramientas y componentes existentes en la UCI o en fase de desarrollo no cuentan con los medios suficientes y necesarios para poder integrarse unas con otras.

Por todo lo expuesto anteriormente se plantea como **problema investigativo** ¿Cómo lograr la interoperabilidad y transferencia segura de datos entre los componentes de una Universidad Virtual para la Universidad de las Ciencias Informáticas (UCI)?

El **objeto de investigación** lo constituye la interoperabilidad y el envío seguro de datos entre herramientas e-learning y como **campo de acción** los estándares para lograr la interoperabilidad y

²⁰ La biblioteca automática es la que cuenta con sistemas de gestión que le permite una ágil y correcta administración de usuarios y de los materiales que resguarda, principalmente en papel.

²¹ Es el acrónimo de Catalogación legible por máquina. Se explicará en el capítulo 1.

²² Se explicará en el Capítulo 1.

transferencia segura de datos entre componentes de una Universidad Virtual para la Universidad de las Ciencias Informáticas.

Como **idea a defender** la interoperabilidad y la transferencia segura de datos entre los componentes de una Universidad Virtual puede alcanzarse a través del uso de estándares.

El **objetivo general** es proponer un modelo²³ para lograr la interoperabilidad y el envío seguro de datos entre componentes de una Universidad Virtual para la Universidad de las Ciencias Informáticas.

Los **objetivos específicos** son:

- Identificar los componentes de una Universidad Virtual para la UCI.
- Analizar los estándares para lograr la interoperabilidad y la transferencia segura de datos entre componentes de una UV.
- Proponer el modelo de interoperabilidad entre los componentes de la UV teniendo en cuenta la transferencia segura de datos.

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

- Identificar y analizar las universidades virtuales con mayor prestigio a nivel mundial.
- Identificar los componentes de la Universidad Virtual.
- Analizar los componentes existentes en la UCI.
- Realizar la propuesta definitiva de los componentes para la UCI.
- Analizar e identificar los estándares de interoperabilidad entre los componentes que fueron identificados para la UV de la UCI.
- Identificar los estándares de seguridad entre herramientas e-learning para el envío seguro de datos.
- Analizar las ventajas que ofrece una arquitectura orientada a servicio para su aplicación en la propuesta de solución.

²³ Un modelo en este trabajo identifica elementos importantes de un contexto o una realidad y establece relaciones entre éstos con el fin de explicar dicha realidad.

- Realizar la propuesta del modelo de interoperabilidad entre los componentes de la UV para la UCI teniendo en cuenta la transferencia segura de datos entre estos componentes.

Los métodos que se utilizan para el desarrollo de la investigación son el **Análisis Histórico – Lógico** para hacer un estudio de cómo ha evolucionado el e-learning y las tendencias actuales de la interoperabilidad entre herramientas que lo soportan, refiriéndose a las UVs, el método **Análisis - Síntesis** para el análisis de la documentación existente relacionada con el tema, extrayendo los elementos más importantes y necesarios para dar solución al problema existente y el método de **Entrevista** para obtener información acerca de las necesidades primarias en una UV y la manera de satisfacerlas, así como consejos y comprensión por parte del entrevistado en pos obtener mejores resultados en el modelo de UV a proponer.

El presente trabajo consta de introducción, dos capítulos, conclusiones generales, recomendaciones, bibliografía y anexos.

En el capítulo 1. “Fundamentos Teóricos” se abordan de forma general los aspectos teóricos más importantes relacionados con el e-learning y sus tendencias actuales. Se realizará una descripción de los componentes de las UVs, los estándares de interoperabilidad teniendo en cuenta aquellos que proveen seguridad. Se realizará un estudio de algunas de las más notables propuestas de UVs a nivel mundial y en Cuba, teniendo en cuenta aspectos relacionados a la interoperabilidad, la seguridad en el envío de datos, así como el uso de estándares.

En el capítulo 2. “Descripción de la solución propuesta” se expone una descripción detallada de la solución propuesta para una mejor comprensión del modelo de interoperabilidad entre los componentes de la UV para la UCI propuesto. Además de algunas acciones a realizar para futuras iteraciones.

CAPÍTULO 1. FUNDAMENTOS TEÓRICOS.

1. Introducción del Capítulo

En este capítulo se realiza un análisis de los principales temas afines a una UV. Se hace un estudio de los principales componentes que definen el marco conceptual de las mismas, tales como: los Sistemas de Administración de Aprendizaje, los Sistemas de Gestión Académicas, Bibliotecas Digitales, Laboratorios Virtuales, Secretarías Virtuales y Repositorios de Objetos de Aprendizaje entre otros.

Se analizan un conjunto de UVs, las cuales constituyen excelentes ejemplos y propuestas en esta modalidad de universidad y que además sirven de referencia en el presente trabajo. Además se hace un estudio de los principales estándares de interoperabilidad que se podrían utilizar para la integración de los componentes de UV tales como: la familia IMS, SQI, OKI, y otros, así como los estándares de seguridad para el intercambio seguro de datos e información entre herramientas e-learning, así como, el provecho de las características y ventajas que ofrece una Arquitectura Orientada a Servicios (SOA) para la interoperabilidad.

1.2. ¿Qué es una Universidad Virtual?

Según bibliografía consultada, muchas son las definiciones de las UVs, a continuación se citan algunos ejemplos:

“...es una forma flexible de enseñanza especializada a distancia que utiliza avanzadas tecnologías de comunicaciones, esto permite ofrecer una formación de excelencia, cumpliendo con los estándares académicos internacionales...” [3]

“...Un Campus Virtual es un espacio digital al que puede acceder la comunidad de una Institución Educativa ya sean profesores/as, alumnos/as o PAS. Una Universidad Virtual sería aquella cuyos servicios son accesibles a través de la red. Paralelamente las Universidades "tradicionales" están

incorporando progresivamente nuevos servicios a su oferta presencial mediante la utilización de Campus Virtuales... [12]

“...es una iniciativa producto de nuestra apuesta por una educación de calidad, abierta e interactiva, donde se rompan las barreras del tiempo y la distancia...” [13]

Partiendo del análisis de los temas estudiados y las definiciones anteriores, se podría conceptualizar a juicio de los autores de este trabajo investigativo que una UV es una forma flexible de enseñanza especializada a distancia que utiliza avanzadas tecnologías de comunicación, esto permite ofrecer una formación de excelencia, cumpliendo con estándares académicos internacionales. Genera además nuevos espacios para que la comunidad universitaria pueda llevar a cabo sus procesos de comunicación, gestión, servicios e investigación.

Como se puede apreciar, existe gran similitud en las definiciones de UVs dadas con anterioridad. Una UV ofrece un ambiente virtual en el que se encuentran representadas todas las actividades docentes y de administración de una universidad tradicional, por tales motivos en la mayoría de los casos se requiere la cooperación e integración de varias herramientas con la responsabilidad de guiar los diferentes procesos dentro de las mismas.

Existe gran uniformidad en la selección de dichas herramientas o componentes, siendo común la utilización de plataformas e-learning como: los LMS o los LCMS para la gestión del aprendizaje del estudiante. Debe aclararse que aunque estos componentes mencionados con anterioridad juegan un rol protagónico dentro del CV, no son los únicos, ni suficientes para catalogarse como UV, siendo este un error común cometido por algunas instituciones que incursionan en la enseñanza a distancia y solo poseen un LMS o LCMS. Éstos y otros componentes que conforman una UV serán descritos a continuación.

1.2.1. Componentes de las Universidades Virtuales

Desde la perspectiva de una UV, el e-learning recoge una amplia gama de utilidades y herramientas que en ocasiones aparecen combinadas formando alguno de los componentes considerables en una UV. Como ha sido mencionado anteriormente, un elemento distintivo dentro del e-learning son las plataformas de aprendizaje, las cuales sirven de soporte a este tipo de enseñanza y no son más que herramientas a través de las cuales se facilita el cursar, la gestión, administración y evaluación de las actividades docentes así como la gestión de los recursos educativos que conforman estas actividades y cursos. Estas plataformas tales como los LMS y LCMS son además una parte modular dentro de las UV, tanto que constituyen uno de sus principales componentes.

Una Secretaría Virtual, una Herramienta de Autor, un Repositorio de Objetos de Aprendizaje, un Sistema de Gestión Académica, un Laboratorio Virtual y una Biblioteca Virtual en agregación de los dos componentes mencionados con anterioridad (LMS y LCMS) son otros componentes que también se integran dentro de una UV y serán descritos en los próximos capítulos.

1.2.1.1. Learning Management System (LMS)

Learning Management System (LMS) son herramientas empleadas en la administración, distribución y control de materiales y actividades didácticas y formativas, gestión de cursos, gestión de clases, herramientas para la comunicación, herramientas para los discentes, gestión del contenido, herramientas de evaluación y gestión de organismos y entidades educativas [14]. Generalmente estos LMS atienden a una estructura modular que permite administrar y configurar diversidad de procesos tales como: la gestión de alumnos y profesores, la creación y calificación de exámenes, la estructuración de cursos, la asignación de los mismos a los estudiantes matriculados y mejoran el aprendizaje del conocimiento y posibilitan que el mismo pueda ser adquirido de una manera centralizada o distribuida.

Según [15], los LMS poseen las siguientes características:

- Perfiles de acceso para ello se definen varios roles (alumno, tutor, profesor, coordinador, administrador), cada uno con diferentes privilegios.

- Las herramientas de comunicación tanto síncronas como asíncronas aspecto fundamental para el aprendizaje y para las relaciones sociales.
- Los servicios y áreas configurables es importante que dependiendo del curso tanto los servicios como las áreas sean configurables.
- La gestión académica y administrativa que permite tener un sistema de gestión de expedientes administrativos, expedientes académicos, control de perfiles de usuarios, administración de cursos, etc.
- Los sistemas de gestión de calificaciones que permiten recoger tanto los resultados obtenidos en pruebas objetivas, como las notas insertadas por los profesores para calificar otras actividades evaluables.

Entre las plataformas LMS más comunes se encuentran Blackboard²⁴, WebCT²⁵, EduStance²⁶, Moodle²⁷ y Claroline²⁸, en el caso de las dos últimas son de código abierto. En los LMS se puede utilizar varias normas o estándares que varía según la plataforma que sea (de los ejemplos antes expuestos), pero en general los comunes en ellas son el SCORM²⁹, IMS CP³⁰, IMS Learning Design (IMS LD³¹), IMS Question and Test Interoperability (IMS QTI³²) y otros.

Cabe destacar, que a pesar de las muchas ventajas y facilidades que los LMS ofrecen, los mismos no pueden crear sus propios contenidos, es decir, no integran en él, medios o herramientas que les permita crear lo contenidos, solo se centran en distribuir y gestionar recursos con fines educativos creados en

²⁴ <http://www.blackboard.com/>

²⁵ <http://www.webct.com/>

²⁶ <http://www.edustance.com/>

²⁷ <http://moodle.org/>

²⁸ <http://www.claroline.net/>

²⁹ Es el estándar con el objetivo de empaquetar los contenidos para importarse y exportarse entre plataformas y compartir, reutilizar, importar y exportar OAs.

³⁰ Es el estándar de empaquetado de contenido propuesto por IMS.

³¹ Es una especificación para describir las estrategias de enseñanza (modelos pedagógicos) y los objetivos educativos.

³² Describe la forma de representar preguntas individuales o ítems y gestionar evaluaciones o exámenes completos.

fuentes externas. Existen otras plataformas que brindan estas facilidades y se focalizan principalmente en la creación de contenidos, tal es el caso de los LCMS (Learning Content Management Systems).

1.2.1.2. Learning Content Management Systems (LCMS)

Los LCMS basan su funcionamiento en los CMS (Content Management System) los mismos tienen por objetivo simplificar la creación y la administración de los contenidos en línea. Los LCMS siguen el concepto básico de los CMS, que es la administración de contenidos, pero orientados al ámbito educativo, administrando y concentrando únicamente recursos con una intención formativa y no todo tipo de información [14].

Así como los LMS, los LCMS garantizan el control y gestión de los procesos de aprendizajes del estudiante por parte del profesor pero desde una óptica diferente y más viable ya que facilitan la creación, almacenamiento y reparto de los contenidos. Con tales fines, los LCMS incorporan una herramienta de autor con el fin de facilitar la creación de OA, los cuales serán a su vez almacenados en el repositorio y un Repositorio de Objetos de Aprendizaje³³ para permitir que los recursos almacenados sean consultados de manera organizada y reutilizados por diferentes usuarios sin que la integridad de la información sea dañada.

El desarrollo de los LCMS es muy lento, principalmente por su complejidad a la hora de implementarlos lo que ha provocado que sean los LMS los más utilizados a nivel mundial.

1.2.1.3. Herramientas de Autor

Como se había mencionado anteriormente, las Herramientas de Autor (HA) se encargan de la creación de OA siguiendo varias normas y rigiéndose por varios estándares, tal es el caso de IMS Content Package (IMS CP) y SCORM estándar mediante el cual es posible empaquetar dichos OA, estas herramientas a su vez emplean los metadatos³⁴ con el objetivo de describir los recursos dentro del objeto, para ello se guía

³³ Se verán más adelante

³⁴ Son un conjunto de atributos o elementos necesarios para describir un recurso.

por el estándar Learning Object Metadata (LOM, que es por el que se basa SCORM para la descripción de los recursos a través de los metadatos), el cual recoge en nueve categorías todos los metadatos con los cuales se manobra en la descripción del OA.

Aunque pocas son las herramientas que trabajan con estos estándares de empaquetado, es válido mencionar a: Reload³⁵, Lompad³⁶, eXeLearning³⁷ y Autore³⁸, figurando como las más utilizadas a nivel mundial en la actualidad, aunque cabe destacar que estas herramientas carecen de la utilización de estándares de interoperabilidad entre sistemas.

1.2.1.4. Repositorio de Objetos de Aprendizajes

En el apartado sobre los LCMS se destaca que gran parte de las ventajas que poseen respecto a los LMS radica no solo en las potencialidades que aprovecha de los CMS, sino de la incorporación además de los ROA, dado que los mismos facilitan la colección o almacenamiento, gestión y reutilización de OA o de información, preservando en todo momento la integridad de los mismos.

Un ROA según [16] “es una colección de OA que tienen información (metadatos) detallada que es accesible vía Internet. Además de alojar los OA los ROA pueden almacenar las ubicaciones de aquellos objetos almacenados en otros sitios, tanto en línea como en ubicaciones locales”.

Ahora, por la forma en la que se concentran los recursos, principalmente se identifican dos tipos de ROA [17] y [18]:

- Los que contienen los objetos de aprendizaje y sus metadatos, en éstos los objetos y sus descriptores se encuentran dentro de un mismo sistema e incluso dentro de un mismo servidor.

³⁵ <http://www.reload.ac.uk/editor.html>

³⁶ <http://sourceforge.net/projects/lompad>

³⁷ <http://exelearning.org/>

³⁸ <http://www.autore.ehu.es/AUTORE>

- Los que contienen sólo los metadatos, en este caso el repositorio contiene sólo los descriptores y se accede al objeto a través de una referencia a su ubicación física que se encuentra en otro sistema o repositorio de objetos.

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

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

En los ROA, existe diversidad en cuanto a los estándares de interoperabilidad de contenido que se utilizan, por ejemplo en algunos para la descripción de los metadatos se hace uso del estándar LOM, además de incorporar estándares para el empaquetado como el SCORM, IMS CP y otros y para la comunicación con otras aplicaciones el IMS DRI, la OSID Repositories de OKI, SQI, etc.

Las UVs, con un ROA como componente pueden almacenar, clasificar, localizar, administrar y recuperar los OA, así como, compartirlos con otros sistemas para diversas aplicaciones, pero esto no es suficiente para apoyar y enriquecer el conocimiento adquirido, es preciso introducir una Biblioteca Virtual (BV) que permita evacuar las necesidades de los estudiantes o profesores referentes al acceso a bibliografías actualizadas, referencias a bases de datos internacionales sobre otros temas de interés que no estén incluidos en los cursos.

1.2.1.5. Bibliotecas Digitales o Virtuales

Las bibliotecas digitales, han permitido a los usuarios tener una disponibilidad a tiempo completo de contenidos digitales que ayuden en su formación permitiendo el acceso universal a los recursos, sin limitantes de tiempo ni espacio, así pueden atender a más usuarios a la vez y gestionar eficientemente

dichos contenidos, como nota importante se debe resaltar que las bibliotecas digitales han sido adoptadas por las bibliotecas tradicionales debido a las muchas ventajas que ofrecen, complementándose así con las mismas.

Por ello, cuando las colecciones de recursos digitales tienen un objetivo claro y se forman con una selección de contenidos organizados con un sistema descriptivo a través de metadatos (catalogación), y además se les asocian algunas facilidades para la búsqueda y uso de la información (servicios), estas colecciones se categorizan como bibliotecas digitales [19].

El contenido almacenado en los repositorios de las bibliotecas digitales es contemplado como Objetos de Información³⁹ que facilitan el acceso a sus contenidos mediante internet y las telecomunicaciones en general. Algunas veces se confunden las bibliotecas digitales con otros tipos de bibliotecas que también hacen uso de la tecnología, pero hay notables diferencias entre unas y otras [20]:

- *Biblioteca automatizada* es aquella que cuenta con sistemas de gestión que le permite una ágil y correcta administración de usuarios y de los materiales que resguarda, principalmente en papel. Asimismo, cuenta con sistemas de telecomunicaciones que permiten acceder a su información de manera remota o local. Proporciona principalmente catálogos y listas de las colecciones que se encuentran físicamente dentro de un edificio.
- *Biblioteca virtual* es aquella que hace uso de la realidad virtual para mostrar una interfaz y emular un ambiente que sitúe al usuario dentro de una biblioteca tradicional. Hace uso de la más alta tecnología multimedia para guiar al usuario a través de diferentes sistemas conectados a través de sistemas de cómputo y telecomunicaciones. Usualmente se requiere de equipos especiales como guantes y cascos para entrar en el espacio virtual que se simula.

Por su parte [21] realizan otra definición de bibliotecas como:

³⁹ Término genérico mediante el cual se puede referir a todo tipo de objeto capaz de proveer información, dígame imágenes, videos, animaciones y multimedios.

- *Biblioteca electrónica* es una expansión de las bibliotecas automatizadas en la que se facilita también el acceso a fuentes o recursos de información electrónica.

Considerando estas definiciones se puede entonces acotar que las bibliotecas en una UV son las Bibliotecas Virtuales pues son las bibliotecas que haciendo uso de las más altas tecnologías de multimedia muestran una interfaz y emulan el ambiente para que el usuario se encuentre como si estuviera en una biblioteca tradicional, pero facilitando además acceso a fuentes o recursos de información electrónica.

En las BV se pueden encontrar estándares como: Circulation Interchange Protocol (NCIP⁴⁰), Z39.50⁴¹ Versión 3, eXtensible Markup Language (XML⁴²), MARC 21⁴³, Dublin Core⁴⁴, The Open Archives Initiative Protocol for Metadata Harvesting (OAI-PMH⁴⁵) y otros.

Las BV como uno de los componentes de las UVs, le brinda la posibilidad a sus usuarios de acceder a información actualizada, pero en las UVs además de poder obtener información que no necesariamente sea referente a los cursos, debe permitirle a sus alumnos adquirir además del conocimiento teórico, un conocimiento práctico y esto se hace a través de los Laboratorios Virtuales.

1.2.1.6. Laboratorios Virtuales

Un Laboratorio Virtual (LV) es un sistema informático que pretende simular el ambiente de un laboratorio real y que mediante simulaciones interactivas permite desarrollar las prácticas de laboratorio que son de gran utilidad pues ayudan al usuario a desarrollar este tipo de habilidad (relacionada con el experimento) a distancia; ayudan en el tratamiento de conceptos básicos, observar, investigar, realizar actividades, así

⁴⁰ Es una norma para la libre circulación de la información y las solicitudes de préstamos inter-bibliotecarios.(Ver <http://ncip.envisionware.com/>)

⁴¹ Se verá en un próximo acápite.

⁴² Lenguaje extendido de marcado(en español)

⁴³ Se verá en un próximo acápite.

⁴⁴ Estándar para la descripción de metadatos en bibliotecas digitales.

⁴⁵ Se verá en un próximo acápite

como también apoyan al estudiante en la elaboración e intercambio de resultados; asumiendo un importante rol en la educación a distancia ya que permite realizar las referidas prácticas de laboratorio desde cualquier computadora fuera y dentro del recinto universitario sin necesidad de cumplir con un horario preestablecido [7].

Estos surgen fundamentalmente, por la necesidad de crear medios de apoyos al estudiante para la realización de las prácticas de laboratorio, con el objetivo de complementar, de una manera óptima el tiempo que éste dedica a la realización de dichas prácticas.

Existen diferentes tipos de laboratorios virtuales [22]:

- I. Laboratorios software.
- II. Laboratorios web.
- III. Laboratorios remotos.

I. Laboratorios software(LS)

Estos laboratorios son programas software que simulan el funcionamiento de un conjunto de instrumentos en los que el alumno sencillamente debe instalarlo en su PC, la cual no necesita conexión a Internet [22].

Por estas características particulares los LS tienen varias desventajas por ejemplo:

- Problemas de versiones.
- Hay ocasiones en que los alumnos no tienen las últimas versiones.
- No existen herramientas colaborativas que permitan el trabajo en grupo.
- No existe posibilidad de que el tutor pueda evaluar de forma continua los progresos realizados por el alumno.
- No trabaja con instrumentos reales.

II. Laboratorios web (LW)

En los LW a diferencia de los LS el alumno se conecta a un servidor web, que es el encargado de servir el programa de simulación, para ello es necesario una PC con conexión a Internet. La principal desventaja de este tipo de laboratorios es que el alumno no trabaja con instrumentos reales y en ocasiones existen problemas con la conectividad al servidor.

III. Laboratorios remotos(LR)

Este laboratorio es basado principalmente en Internet, pues el alumno se conecta a un servidor web que le mostrará las imágenes reales de los instrumentos que va a manejar, las acciones que puede realizar y los resultados de esas acciones. La principal desventaja es la necesidad de una buena conexión a Internet.

Los LV recomendados a utilizar en las UVs son los laboratorios remotos porque a pesar de que se tiene que contar con una buena conexión a internet borra las barreras del tiempo y el espacio, que es el principal objetivo de las UVs, sin obviar que los mismos pueden intercambiar información con otras aplicaciones como un LMS, utilizando el estándar IMS-QTI⁴⁶, por mencionar uno.

En general en los LV se evalúan los conocimientos teóricos adquiridos, a través de prácticas de laboratorios, pero estas evaluaciones deben ser registradas en las UVs, siendo este uno de los tantos motivos por los cuales se incorpora otro componente como los Sistemas Gestión Académicos que no solo brindan estas funcionalidades, sino también otras que serán vistas en el próximo acápite.

1.2.1.7. Sistema de Gestión Académica (SGA)

En la actualidad existe una gran variedad de conceptos o definiciones acerca de los SGA, en los que se destacan los siguientes:

⁴⁶ Se explicará en un próximo acápite.

“...es una solución de gestión y de análisis que agrupe toda la información relacionada con la labor académica y que, por tanto, incluya todos los procesos de la institución. “ Al mismo tiempo, todos estos procesos están integrados con el resto de sistemas de información de la organización, es decir, con la gestión de recursos humanos y la económica” [23].

“...es un sistema de gestión de alumnos que registra y administra todas las actividades académicas de la universidad, desde que los alumnos ingresan como aspirantes hasta que obtienen el diploma. Pasando por un proceso de matriculación, el registro de cursado de materias y de resultados académicos, los pedidos de equivalencia y la gestión del egresado...” [24].

Luego de ver analizado los temas de SGA y las definiciones anteriores se podría conceptualizar que:

Un Sistema de Gestión Académicas (SGA) es una herramienta que les permita a la universidad administrar la gestión de alumnos en forma segura, con la finalidad de obtener información consistente para los niveles operativos y directivos y que además proveen mecanismos automatizados que facilitan la planificación, organización, gestión y control académico, brindando servicios a los alumnos, docentes, personal administrativos y autoridades de todo el sistema universitario.

Los SGA para el intercambio de información con otras aplicaciones utilizan varios estándares encaminados a favorecer la interoperabilidad, como son: el IMS Learner Information Package (IMS LIP⁴⁷), IMS Enterprise Service (IMS ES⁴⁸), LTSC PAPI⁴⁹, etc.

Estos SGA permiten administrar y registrar todas las actividades académicas de la universidad, a lo largo de todo el proceso, desde que el estudiante ingresa como aspirante hasta que obtiene el diploma, incluyendo en algunos casos los procesos de matriculación en la UV, pero en el caso en que el SGA implantado no lo contemple, es necesario la implantación de una Secretaría Virtual, para realizar esta

⁴⁷ Se explicará en un próximo acápite.

⁴⁸ Se analizará más adelante.

⁴⁹ Learning Technology Standardization Committee (LTSC) Public and Private Information (PAPI) es el estándar que se dedica a apoyar el intercambio de datos de alumnos entre diferentes sistemas. Se especifica tanto la sintaxis y la semántica de un "Modelo de Aprendizaje", que caracterizan a un alumno y sus conocimientos y habilidades.

operación, llevar a cabo procesos de gestión económica, de carácter administrativo y otras actividades afines a una SV.

1.2.1.8. Secretaría Virtual

La Secretaría Virtual (SV) es la web para la gestión on-line y/o descarga de impresos de solicitudes para servicios de informática. A través de las SV se puede consultar los expedientes académicos, calificaciones, estado de la solicitud de beca, estado de la solicitud de título y el estado de su matrícula y además se puede realizar el pago online de la matrícula a determinado curso [25].

En una SV son brindadas y convergen una serie de facilidades y funcionalidades en general que responden a las necesidades de gestión e información por parte de estudiantes y de la comunidad universitaria. Una de las principales funcionalidades en esta modalidad de secretarías recae sobre la *búsqueda de información*, la cual se concibe lo más flexible y exhaustiva posible de modo tal que el estudiante tenga en tiempo real la información correcta y más actualizada sobre sus exámenes, plan de estudio, cursos y otras noticias y eventos, etc.

Otras de las funciones de una SV está ligada a la *realización de trámites académicos*, entiéndase por ello a la realización de cualquiera de las gestiones habituales en una universidad tales como matricularse, solicitar convalidaciones, la expedición de documentos oficiales (certificados, etc), becas, ayudas al estudio, confección de sus calendarios de prueba , etc.

Otro de los elementos básicos de una SV y de gran valor tanto por la información que contiene, como por la utilidad que tiene para la comunidad universitaria, particularmente para los estudiantes y sus tutores. Se trata del *expediente académico*. En él pueden recogerse diversos tipos de información, desde la más clásica, relativa a las calificaciones del estudiante, hasta la más innovadora, relativa al estado de las solicitudes realizadas por el estudiante o al efecto que la aplicación de la normativa universitaria haya podido producir en el expediente académico, y todo ello en tiempo real. Así mismo, se puede disponer de la información sobre la resolución de las convalidaciones que se han solicitado, las sanciones que pueda tener su expediente, las autorizaciones que se hayan producido, su duración, etc. Este suele ser uno de

los servicios más solicitados dentro de una SV ya que no solo permite a un estudiante saber del estado de sus expediente sino que permite a profesores y tutores adquirir información muy importante a la hora de valorar la situación del estudiante [25].

Por último, la secretaría puede disponer de otros elementos accesorios que facilitan la difusión transversal de la información académica, tales como el *calendario académico*, desde donde se recuerdan las fechas clave de cada mes y se facilita el acceso directo a la información correspondiente y a la solicitud de los trámites que corresponda, y el tablón o pizarra virtual de anuncios, una reproducción virtual de lo que vendría a ser el tradicional tablón de anuncios de cualquier secretaría destinado a recoger aquellas informaciones generales de interés. Es por ello que existe una estrecha relación entre la SV y el SGA.

De forma general son varios los estándares que son utilizados en las SVs tales como: el IMS Learner Information Package (IMS LIP), IMS Enterprise Service (IMS ES), LTSC PAPI y otros.

En resumen estos son algunos de los componentes que contienen las UVs estudiadas, en el próximo apéndice se ejemplifican algunas de ellas.

1.2.2. Ejemplos de Universidades Virtuales

Son disímiles las propuestas de UVs en todos los continentes, en las próximas acápite se tratarán aspectos de tres Universidades que han encaminado esfuerzos en el desarrollo de la educación a distancia, que son destacadas en el Ranking Web de Universidades del Mundo por sus resultados, las excelentes críticas recibidas así como su posicionamiento en internet, incorporando además la Universidad Virtual del CITMA en Cuba por sus esfuerzos en la integración de sus componentes y por ser una de las propuestas que más se acerca a las UVs a nivel mundial en el país.

1.2.2.1. Universidad Oberta de Cataluña (UOC)

Nacida como UV en Barcelona en el año 1994, la Universidad Oberta de Cataluña (UOC) ofrece un CV que permite la comunicación de toda la comunidad universitaria, estudiantes, profesores y personal de

gestión y se pueden ver reproducidos en él, los servicios de cualquier campus universitario tradicional, pero adaptados a un entorno virtual, en el que, a su vez, desaparecen la mayoría de las limitaciones impuestas por horarios, distancias, etc.

En el CV de la UOC se incluye una BV para consultar información relacionada con los cursos que se imparten y documentación de otros aspectos que no sean docentes, utilizando el Campo⁵⁰ 856 que se encuentra dentro de los formatos LCMARC⁵¹ para la catalogación de documentos electrónicos, el formato Tagged Image File Format (TIFF⁵²) para el envío a los usuarios de las peticiones de documentos, etc.

Además, contiene una SV a través de la cual se pueden realizar trámites académicos, tales como: matricularse, solicitar convalidaciones, la expedición de documentos, becas y ayudas al estudio, así como configurarse su propio calendario de exámenes, eligiendo los días, las horas y el lugar que más le convengan para la realización de los mismos, etc. , contiene el catálogo de servicios en el que puede encontrar y, sobre todo, identificar, el servicio que precisa, así como las informaciones adicionales inherentes al mismo, tales como: si está sujeto al pago de una tasa o no, si se necesita alguna autorización especial, si se generará una resolución formal, etc.

Además la UOC cuenta con un sistema de gestión académica telemática (GAT), que permite gestionar los procesos académico-administrativos relacionados con los estudiantes, los docentes y los usuarios de administración, utilizando para el intercambio de información de los estudiantes con el LMS los estándar IMS LIP e IMS Enterprise Information Model (IMS EIM⁵³). Asimismo, para todo el proceso de gestión del conocimiento utiliza el LMS Moodle y un ROA, utilizando para el intercambio de contenido el estándar SCORM que le permite a la plataforma importar, exportar y visualizar materiales de aprendizaje que cumplan con este estándar.

⁵⁰ Una sección del documento.

⁵¹ Denominación inicial del estándar MARC 21.

⁵² Formato de archivo de imágenes con etiquetas que describen las imágenes ("*Tagged Image File Format*").

⁵³ Es una especificación que define la interoperabilidad de sistemas que residen dentro de la misma organización.

La UOC basa la interoperabilidad entre sus componentes tomando como referencia el proyecto CAMPUS⁵⁴, el cual define una estructura de integración en la que intervienen entre otros elementos; una pasarela o gateway y un Bus de OKIS⁵⁵, dicho Bus permite usar en forma de web services los OSIDs⁵⁶ básicas de OKI (autenticación, autorización, seguimiento y configuración) para conectar una aplicación. Este Bus demanda además la existencia de un LMS (Moodle, Sakai, etc.), así como una pasarela que permita el entendimiento entre el Bus de OKI y el LMS correspondiente⁵⁷.

Entre las disímiles ventajas que se han mencionado de la UOC hay que destacar una de sus desventajas radica en los precios para obtener los cursos, pues es sumamente caro, sin incluir que los libros que complementan los estudios en la UOC no se incluyen dentro del precio de la matrícula y para acceder a ellos hay que comprarlos.

1.2.2.2. UNED de España

La Universidad Nacional de Educación a Distancia (UNED) de España fue fundada en 1972 tras la idea de la Universidad Abierta del Reino Unido para garantizar la calidad de la educación superior a través del sistema de educación a distancia.

Una de las características de la UNED es que imparte cursos a distancia por medio de apoyo audiovisual e internet y contiene un CV donde se le ofrece al estudiante todos los servicios necesarios para su desarrollo en el proceso de aprendizaje, permitiendo a los alumnos tener acceso al material docente, BV, foros y debates, tablón de noticias, gestión de documentos y de proyectos, charlas, agenda, enlaces favoritos y correo electrónico, entre otros medios pedagógicos. Para poder realizar el proceso de aprendizaje a distancia en el CV de la UNED se utiliza la plataforma de aprendizaje e-learning ALF⁵⁸,

⁵⁴ Ver <http://www.campusproject.org> o <http://www.campus.cat>

⁵⁵ Se verá más adelante

⁵⁶ Se explicará más adelante

⁵⁷ Para más información consultar <http://www.campusproject.org>.

⁵⁸ <http://www.innova.uned.es/index>

desarrollada por la propia universidad, que permite enviar y recibir información, gestionar y compartir documentos, crear y participar en comunidades temáticas, etc.

En este CV por la diversidad de sus alumnos, se realiza la comunicación entre sus componentes a través del RDSI⁵⁹ (Red Digital de Servicios Integrados) y el RTC⁶⁰ (Red Telefónica Conmutada). Además contiene un Sistema de Tele-Presencia, el cual permite a los usuarios recibir y transmitir video y/o audio captada por una cámara, utilizando la norma H.323⁶¹ para la comunicación punto a punto [26].

Además tiene una SV para realizar los procesos relacionados con trámites administrativos como: consulta de matrícula, estado de la matrícula, el estado de la solicitud de beca y acceso al expediente académico, así como el pago de las matrículas de los cursos. Incoepora un Repositorio digital en el que se puede localizar la producción científica de la universidad a texto completo, siendo éste el espacio de los contenidos digitales de la UNED, que utiliza el middleware de integración Fedora⁶² con arquitectura SOA y maneja para el control de acceso al mismo, el estándar XACML⁶³ y como el formato de empaquetado de los OA el estándar SCORM.

La UNED además contiene un conjunto de LV de diferentes asignaturas para poner en práctica los conocimientos teóricos adquiridos en los cursos, integrándolo con el LMS a través de un middleware capaz de utilizar los diferentes servicios ofrecidos por el LMS(seguimiento de alumnos, gestión de contenidos, evaluación, etc.), los estándares e-learning (SCORM, IMS-QTI, etc) y el componente práctico que aportan los laboratorios virtuales [22], un ejemplo de laboratorio en la UNED es el “Laboratorio Virtual y Remoto” para los estudiantes que están matriculado en la asignatura de Automática I de la especialidad de Física Industrial en la carrera de Ciencias Físicas de la UNED.

⁵⁹Facilita conexiones digitales extremo a extremo para proporcionar una amplia gama de servicios, tanto de voz como de otros tipos, y a la que los usuarios acceden a través de un conjunto de interfaces normalizados

⁶⁰ Es una red de comunicación diseñada para la transmisión de voz, aunque pueda también transportar datos.

⁶¹ Es una familia de estándares definidos por el ITU para las comunicaciones multimedia sobre redes LAN, siendo una tecnología de principio básico para la transmisión en tiempo real de audio, vídeo, y comunicación de datos punto a punto.

⁶² <http://www.fedora-commons.org/>

⁶³ Se verá más adelante

También cuenta con una Biblioteca para acceder a información de interés del usuario e información acerca de la distribución de los libros, sus números y las colecciones especiales, donde se ofrece un catálogo, los libros electrónicos, acceso a las bases de datos, tesis doctorales, revistas, recursos de Internet, etc. La Biblioteca de la UNED utiliza el Sistema Integrado de Gestión de Biblioteca, UNICORN⁶⁴ el cual está diseñado para soportar estándares como: Circulation Interchange Protocol (NCIP⁶⁵), Z39.50⁶⁶, eXtensible Markup Language (XML⁶⁷), The Open Archives Initiative Protocol for Metadata Harvesting (OAI-PMH⁶⁸), MARC 21⁶⁹, Dublin Core⁷⁰ y otros.

La UNED de España presenta demoras en las evaluaciones, las herramientas docentes virtuales están infrautilizadas y el campus virtual no es muy usable pues es lento y engorroso [27].

1.2.2.3. UNED de Costa Rica

La Universidad Estatal a Distancia de Costa Rica (UNED) es otra propuesta de UV pero en la región latinoamericana, fue creada por Ley de la República, el 22 de febrero de 1977, desde su creación se ha caracterizado por la producción académica de calidad; la investigación en el campo del uso de los recursos tecnológicos y en los procesos de inducción y acompañamiento a los estudiantes.

Desde el año 1995 incursionó en el uso de aplicaciones tecnológicas para la virtualización de procesos académicos. El uso de plataformas de aprendizaje en línea es complementado con otros recursos como: videoconferencias, audiovisuales, multimedia o material impreso.

⁶⁴ Ver <http://www.sirsi.es/unicorn.html>

⁶⁵ Es una norma para la libre circulación de la información y las solicitudes de préstamos inter-bibliotecarios.(Ver <http://ncip.envisionware.com/>)

⁶⁶ Se verá en un próximo acápite.

⁶⁷ Lenguaje extendido de marcado(en español)

⁶⁸ Se verá en un próximo acápite

⁶⁹ Se verá en un próximo acápite.

⁷⁰ Estándar para la descripción de metadatos en bibliotecas digitales.

En la UNED de Costa Rica, los procesos de virtualización comenzaron con la plataforma Microcampus, que es un LMS creado fundamentalmente por la Universidad de Alicante, pero éste no era suficiente para cubrir todas las necesidades de la universidad, luego se decide la transformación de la Unidad de Microcampus (que era un analista y un administrador para capacitar operativamente a los profesores en el uso de la plataforma y cómo administrarla) en lo que se conoce como el Programa de Aprendizaje en Línea(PAL), junto con esto se adquiere la plataforma WebCT y da inicio a los procesos de desarrollo y pilotaje de la plataforma de fuente abierta Moodle.

Además la UNED cuenta con una BV en la que se puede encontrar los reglamentos a texto completo, catálogos para acceder a la información tanto académica como institucional y el desglose del sistema de bibliotecas, utilizando para el intercambio de datos el estándar Extended Markup Language (XML). Posee un conjunto de LV que dan soporte y ponen en práctica los conocimientos de varias asignaturas que se imparten en la misma.

La UNED de Costa Rica presenta varias desventajas, algunas de ellas están referida a los LVs que se utilizan, los cuales son del tipo Laboratorios Software los cuales se les entrega en un CD a los estudiantes para realizar sus prácticas, pero para evaluarse tienen que dirigirse a las sedes de la misma, no existe posibilidad de que el tutor pueda evaluar de forma continua los progresos realizados por el alumno, no existen herramientas colaborativas que permitan el trabajo en grupo y además se ve un poco engorroso el uso de tantos LMS (Microcampus, WebCT, Moodle y PAL) para el aprendizaje.

1.2.2.4. Universidad Virtual del Ministerio de Ciencia, Tecnología y Medio Ambiente (CITMATEL)

CITMATEL es la Universidad Virtual del Ministerio de Ciencia, Tecnología y Medio Ambiente (CITMA) de Cuba, destinada a la educación a distancia, donde se integran aulas, bibliotecas, eventos, y museos virtuales, así como ambientes de trabajo colaborativo. El ambiente del aula virtual se presenta sobre páginas web con la guía metodológica y sirve de base para contenidos complementarios particulares.

Para la descripción de los recursos de información gestionados dentro del sistema, fue utilizado Dublin Core con el objetivo de ayudar en su organización y Recuperación [28].

Además en la Universidad Virtual del CITMA se trabajan con los laboratorios virtuales de Física y Matemáticas y tiene un Sistema de Gestión del Aprendizaje que incorpora servicios de comunicación sincrónica y asincrónica.

Los servicios que brinda CITMATEL constituyen una opción para facilitar la instrucción y el adiestramiento, están basados en la aplicación de las TIC, que permiten el aprendizaje sin las limitaciones del lugar, tiempo, ocupación o edad de los estudiantes que a ella se matriculen. Se ofrece con carácter limitado a organizaciones o personas a través de una clave de acceso obtenida al matricular a uno de los cursos. Para el proceso de matriculación de los cursos y pago de los mismos, CITMATEL remite el pago a través de la tienda virtual " bazar virtual"⁷¹ en la sección referente a los Cursos en Línea.

Esta universidad aún es muy insipiente pues no contempla componentes como los sistemas de gestión académicos y además contiene una cantidad de usuarios restringida por el carácter limitado de matriculación a organizaciones o personas a través de una clave de acceso obtenida al matricular a uno de los cursos y por el pago de los cursos es con moneda convertible.

Para lograr una transferencia e intercambio efectivo de información entre los componentes de una UV se requiere el uso estándares. A continuación serán tratados temas referentes a estándares, en especial los de interoperabilidad entre aplicaciones y seguridad.

1.3. ¿Qué es un estándar?

El estándar es definido por diferentes autores de varias formas, pero todas en esencia se basan en un patrón, una tipificación o una norma de cómo realizar algo [12], que es de obligado cumplimiento en un entorno determinado, pero antes de establecer un estándar primero se realizan las especificaciones que

⁷¹ <http://www.bazar-virtual.com/>

son las descripciones detallada y completas de las características, naturaleza o forma de realizarse un objeto o procedimiento para obtener un resultado deseable.

Con los estándares se puede garantizar que existan normas que rijan el intercambio de información, a estos, se les conoce como estándar de interoperabilidad.

1.3.1. Estándares de interoperabilidad entre los componentes de las UV

En la actualidad son disímiles los esfuerzos realizados por diferentes instituciones que desarrollan especificaciones y estándares orientados hacia una forma común de identificar, definir y comunicar a todos los recursos involucrados en un entorno e-learning, las más reconocidas son:

- **AICC⁷²(Aviation Industry Computer-Based Training Comitee)**. Es una asociación de entrenamiento profesional basado en tecnología, especializado en el sector de la aviación pero que se ha permeado también a otros sectores. Se reconoce como una de los precursores de la estandarización de materiales del entrenamiento profesional.
- **IMS Global Consortium Inc.⁷³**. Cuenta con miembros de organizaciones comerciales, educativas y gubernamentales dedicadas a definir y distribuir arquitecturas abiertas para actividades de educación en línea. Uno de sus resultados es lo que se conoce como el estándar IMS y sus especificaciones
- **Advanced Distributed Learning (ADL⁷⁴)**. El ADL es una iniciativa lanzada por Departamento de Defensa de Estados Unidos y la Oficina de Ciencia y Políticas Tecnológicas de la Casa Blanca en 1997 que permite proveer acceso de la más alta calidad en educación y entrenamiento, en cualquier lugar y en cualquier momento creando el modelo SCORM.
- **ARIADNE⁷⁵(Alliance of Remote Instructional Authoring and Distribution Networks for Europe)**. Es un proyecto de investigación y de desarrollo tecnológico de telemática para la educación y el entrenamiento, patrocinado por la Unión Europea. El proyecto se enfoca al desarrollo de herramientas

72 <http://www.aicc.org/>

73 <http://www.imsproject.org/>

74 <http://www.adlnet.org/>

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

y metodologías para producir, administrar y reutilizar elementos pedagógicos basados en computadora, así como el currículo de entrenamiento a distancia.

- **MIT (Instituto Tecnológico de Massachusetts)** es una de las principales instituciones dedicadas a la docencia y a la investigación en Estados Unidos.
- **Prolearn.** Es una red de excelencia financiada que tiene como misión reunir a los más importantes grupos de investigación en el área de aprendizaje profesional y la formación, así como otras organizaciones clave y socios industriales.
- **IEEE/LTSC⁷⁶(Institute of Electrical and Electronics Engineers/Learning Technology Standards Committee).** El IEEE es una asociación internacional, cuya misión es promover los procesos ingenieriles para la creación, desarrollo, integración, compartición y aplicación del conocimiento sobre tecnologías electrónicas y de información. Dentro de su organización cuenta con el Comité de Estándares para Tecnología del Aprendizaje o LTSC, que se encarga de desarrollar estándares técnicos, recomendaciones y guías para la tecnología educativa.
- **W3C⁷⁷(World Wide Web Consortium).** El consorcio W3C se encarga del desarrollo de tecnologías interoperables (especificaciones, normas, software y herramientas) para aprovechar todo el potencial del Web. Aunque este consorcio no está directamente vinculado con el desarrollo del e-learning es importante mencionarlo ya que de la interoperabilidad de la web dependen muchas de las funciones de la educación en línea.

Todos estos grupos de desarrollo han llevado adelante la estandarización del e-learning de un modo o de otro, pero las más reconocidas y aplicadas son las propuestas de IMS Learning Consortium, la de ADL, el MIT y Prolearn, que han tenido como resultado las especificaciones IMS, el modelo SCORM, el estándar OKI junto a las OSID y el estándar SQL, respectivamente [4].

En las siguientes secciones se describen con más detalle éstos y otros estándares de interoperabilidad pero antes de proseguir con la descripción de los mismos debe aclararse que a pesar de haber dedicado esta sección al tratamiento y descripción de aquellos estándares que posibilitan la interoperabilidad entre diversas aplicaciones y sistemas, se ha decidido incorporar el estándares SCORM ya que establece normas

⁷⁶ <http://www.ieee.org>

⁷⁷ <http://www.w3.org>

de empaquetado que permiten la reutilización de los contenidos en diversos contextos y por diferentes herramientas. Se incluyen además los estándares: Marc 21 e ISO 2709 para la catalogación de recursos bibliográficos.

1.3.1.1. Shareable Content Object Reference Model (SCORM)

SCORM (*Shareable Content Object Reference Model*) es una iniciativa de ADL que consiste en un conjunto de especificaciones y estándares, elaborados por distintos grupos o instituciones (IEEE, AICC e IMS) que se postula como el modelo común para los OA. Este está diseñado para compartir, reutilizar, importar y exportar OA, además describe cómo las unidades de contenidos se relacionan unas con otras a diferentes niveles de granularidad, cómo se comunican los contenidos con el LMS, define cómo empaquetar los contenidos para importarse y exportarse entre plataformas, y describe las reglas que un LMS debe seguir a fin de presentar un aprendizaje específico.

SCORM está encaminado a permitir que se compartan contenidos educativos estándar entre diferentes sistemas, facilitando la interoperabilidad y potenciando la reutilización de los contenidos educativos, para ello define esencialmente dos libros a partir de las versiones 1.2 y un libro adicional (Secuenciamiento y navegación de contenidos (SN - *Sequencing and Navigation*) a partir de la versión de 2004 [29]:

- **Agregación de contenidos (CAM–Content Aggregation Model):** Este modelo brinda la posibilidad, a través de una descripción, de conocer cómo deben ser descritos los OA para facilitar su búsqueda y localización (proponiendo utilizar el estándar de metadatos IEEE LOM⁷⁸), define cómo agruparlos y empaquetarlos para crear unidades más complejas, que se puedan transportar entre diferentes sistemas y además, especificar las reglas para establecer una secuencia de OA que conformen unidades más complejas (tales como cursos).
- **Entorno de ejecución (RTE – Run-Time Environment):** En éste se describe cómo se debe realizar el proceso de ejecución de los contenidos, es decir, que se pueda visualizar, escuchar, obtener, etc. de manera correcta el OA, en la computadora del usuario final y define un modelo de datos estándar

⁷⁸ IEEE LOM (*Learning Object Metadata*) es el estándar de metadatos para objetos de aprendizaje elaborado por IEEE

con el que se pueda obtener información significativa sobre el alumno y la experiencia educativa que lleva a cabo mientras utiliza los contenidos.

- **Secuenciamiento y navegación de contenidos (SN - *Sequencing and Navigation*):** En el modelo SN se define cómo interpretar las reglas de secuenciación de los contenidos y la secuencia de contenidos para un usuario.

Para el funcionamiento del SCORM todos los contenidos se convierten en paquetes SCORM que además del contenido, este paquete estará formado por los metadatos y una lista completa de los contenidos del mismo.

En la figura 1 se muestra la estructura de un paquete SCORM el cual contiene un fichero xml denominado “manifiesto” y es el encargado de recoger la información del paquete así como la estructura organizacional del paquete a generar. Este archivo xml se divide principalmente en cuatro secciones fundamentales según [29]:

- **Meta-data:** metadatos del paquete.
- **Organizations:** Contiene uno o varios elementos <organization> que pueden reflejar diferentes estructuras para los mismos contenidos (secuenciales, jerárquicas, etc.)
- **Resources:** Ficheros físicos almacenados en el manifiesto (que pueden ser assets o SCOs) y referencias a recursos externos.
- **Sub-Manifest(s):** se utilizan cuando existe agregación de contenidos en un mismo paquete, es decir, existencia de unos paquetes dentro de otros.

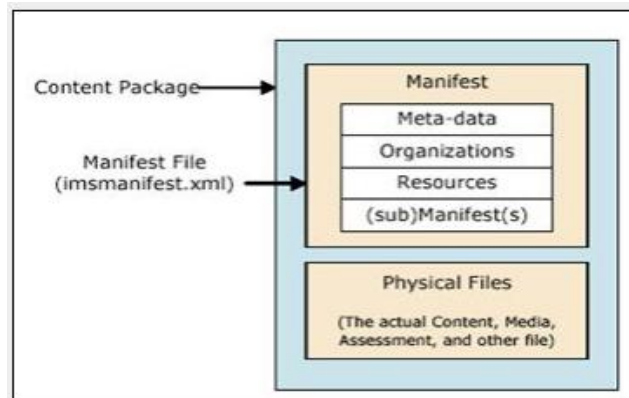


Figura 1 Un paquete de contenidos SCORM [29]

Para la solución se propone utilizar el estándar SCORM porque además de proporcionar la funcionalidad para describir y empaquetar materiales de aprendizaje, tales como cursos individuales o una colección de cursos, en paquetes interoperables y distribuibles, también direcciona la descripción, estructura y ubicación de materiales de aprendizaje en línea, así como la definición de algunos tipos específicos de contenidos y asegura la compatibilidad, la importación y la exportación con cualquier herramienta que soporte esta especificación junto con la de IMS, pues esta engloba la especificación de empaquetamiento de contenidos IMS CP⁷⁹.

1.3.1.2. MACHINE-Readable Cataloging (MARC 21)

MARC (MACHINE-Readable Cataloging) es un registro catalográfico legible por máquina, es decir, es un registro bibliográfico que contiene la información que tradicionalmente se presenta en una ficha de catálogo de biblioteca donde un tipo particular de máquina o computadora, puede leer e interpretar los datos contenidos en ese registro [30].

⁷⁹ Especificación del estándar IMS para describir y empaquetar contenidos de forma que puedan ser procesados por otro LMS diferente.

El objetivo principal es permitir el intercambio de datos, o importar datos de diferentes instituciones o exportar datos de su institución para otros sistemas o redes de bibliotecas a través de los programas de ordenador creado específicamente para ello.

Asimismo la Biblioteca del Congreso de Washington que sirve como repositorio oficial de las publicaciones de los Estados Unidos de América desarrolló el Formato LC MARC, como un sistema de aplicación de números, letras y símbolos en registros catalográficos que permitiera marcar diversos tipos de información, luego este el formato original LCMARC se transformó en MARC 21 [30].

MARC 21 es basado en la norma ISO 2709⁸⁰ y ha sido diseñado para servir como portador de la información bibliográfica relativa a: materiales textuales impresos y manuscritos, archivos de computador, mapas, música, recursos continuos, materiales visuales y materiales mixtos. La información bibliográfica comúnmente incluye: títulos, nombres, tópicos, notas, datos de publicación e información sobre la descripción física de un ítem o elemento.

1.3.1.3. Instructional Management System (IMS)

El estándar IMS es el resultado de varios esfuerzos de la IMS Global Consortium Inc, para facilitar las actividades de aprendizaje sobre tecnología web, principalmente para el intercambio de contenidos y de información sobre los estudiantes. IMS tiene muchas especificaciones ya que cada una de ellas está enfocada en una necesidad distinta del proceso de enseñanza. Las especificaciones IMS son el resultado de una activa iniciativa que está desarrollando y proponiendo especificaciones basadas en tecnologías abiertas (XML), que cubre, entre otras facilidades, accesibilidad y adaptación del estudiante, la definición de competencias, el empaquetamiento de contenidos, información de agentes del proceso educativo, el diseño del aprendizaje a través de un lenguaje para expresar diferentes modelos pedagógicos, así como la formación de repositorios de contenidos digitales.

Todas las especificaciones de IMS cuentan con al menos tres documentos [14]:

⁸⁰ Es un estándar de la Organización Internacional para la Estandarización (ISO) destinado a la descripción bibliográfica.

Information Model (Modelo de Información). Describe de manera conceptual la estructura de los datos, elementos y demás componentes que dan cuerpo a la especificación.

Best Practice and Implementation Guide (Guía de buenas prácticas y de implementación). Explica cómo implementar una especificación IMS a un sistema específico, orienta al equipo de desarrollo durante el proceso de implementación.

XML Binding (Ligadura XML). Este documento es una guía de notas e información sobre cómo representar el Modelo de Información a través de elementos XML, explicando las convenciones de los esquemas XML que se utilizan e incluyendo el código.

A continuación se describen en detalle algunas de las especificaciones consideradas para este trabajo.

1.3.1.3.1. IMS Digital Repositories Interoperability (IMS-DRI)

Esta especificación provee recomendaciones para la interoperabilidad de las funciones más comunes entre repositorios. En el nivel más general, define los repositorios digitales como colecciones de recursos con acceso a través de una red, sin conocimiento previo de la estructura de la colección. Los repositorios pueden contener los objetos o los metadatos que los describen y no importa si los objetos y los metadatos se encuentran en diferentes repositorios.

Esta especificación se propone para la interoperabilidad entre servicios o aplicaciones que tienen las funciones más comunes de un repositorio: buscar, exponer, coleccionar, enviar, almacenar, pedir, entregar y alertar. Entre estas funciones, se reconocen cinco combinaciones como actividades principales las cuales se explican en la siguiente tabla [4]:

Función	Descripción	Recomendación tecnológica
Buscar/Exponer (Search/Expose)	Ejecuta la búsqueda de metadatos asociados con los contenidos que el repositorio expone.	Se recomienda XQuery para colecciones con metadatos en XML y Z39.50 para búsquedas en sistemas bibliotecarios.
Colectar/Exponer (Gather/Expose)	Define la solicitud de metadatos que el repositorio expone, la	No hay recomendación específica, pero IMS DRI sugiere que OAI puede proveer

	agregación de los metadatos para utilizarse en búsquedas subsecuentes y la agregación de metadatos para crear nuevos repositorios.	una funcionalidad adecuada.
Enviar/Almacenar (Submit/Store)	Se enfoca a la manera en la que un objeto se mueve a un repositorio desde un sitio accesible por red y cómo el objeto será representado en el repositorio para su acceso.	Se recomienda el uso de paquetes IMS a través de Simple Object Access Protocol (SOAP ⁸¹).
Pedir/Entregar (Request/Deliver)	Permite, que una vez que el usuario a localizado los metadatos en la función Buscar, pueda solicitar al repositorio el acceso al recurso.	Recomendación general para utilizar HTTP y FTP, para diferentes tipos de recursos. También IMS CP.
Alertar/Exponer (Alert/Expose)	Función clave, en la que a través de correo electrónico se notifica a los usuarios sobre eventos en el repositorio, pero no está contemplada todavía en esta versión de la especificación.	No hay recomendación específica ya que esta función sale del alcance de esta especificación.

Tabla 1. Descripción y recomendaciones tecnológicas para las funciones de un repositorio

En la figura 2 se exhibe una arquitectura funcional que toma como partida las funciones vistas en la tabla 1. Además se muestran los roles, los componentes funcionales y los servicios que definen el espacio de

⁸¹ Es el protocolo de comunicación para los Servicios Web.

interacción completo, pero IMS DRI no incluye a todas éstas, sólo se enfoca a la interoperabilidad de las funciones que se han mencionado. (Ver Figura 2)

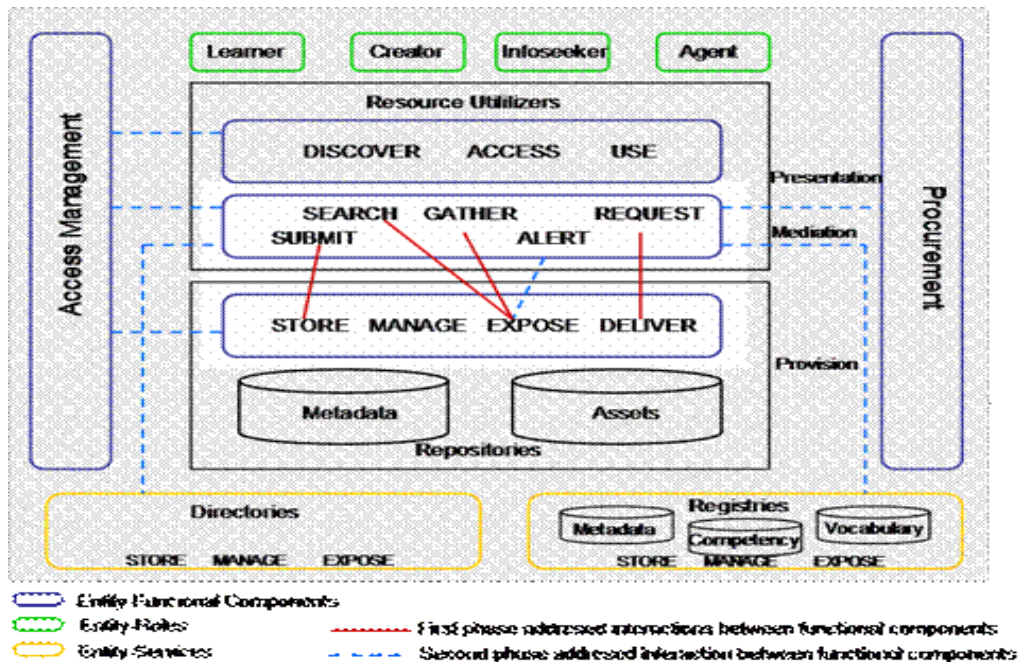


Figura 2. Arquitectura funcional. Reproducida de IMS DRI Information Model [31]

Este estándar como se expuso en el “Framework Integrador para la interoperabilidad entre herramientas que gestionan contenidos educativos reutilizables” por [4], será uno de los utilizados en la solución de este trabajo, debido a que define cuáles son las funciones necesarias para que los repositorios se comuniquen entre sí, aunque en algunos casos no serán utilizados los estándares que recomienda la especificación para cada función.

1.3.1.3.2. IMS Enterprise Services

IMS Enterprise Services es un estándar internacional de formato de archivo XML que puede utilizarse para especificar matrículas / anulaciones en los cursos, información del curso, información de usuario, mantenimiento de expedientes y administración de grupos desde el punto de vista administrativo [15].

Además es la definición de cómo gestionar el intercambio de información que describe personas, grupos y compañías en el contexto de aprendizaje. La especificación Enterprise Services se basa en los conceptos de [32]:

Interoperabilidad: Esencialmente para centrarse en el intercambio de información entre los sistemas de la empresa.

Orientada a Servicios: Esta especificación define el intercambio de información en términos de servicios que se suministran por la colaboración de otros sistemas.

Basado en Componentes: En este el conjunto de los servicios serán ofrecidos de manera que se pueden combinar para formar una gama de servicios.

Múltiples Enlaces – El modelo de información de Enterprise Services se define utilizando el Lenguaje Unificado de Modelado (UML). Esto posibilita el mapeo de la información en el modelo a través de diferentes enlaces. Los enlaces son de importancia inmediata para el Lenguaje de descripción de servicios Web. (WSDL).

Adopción - Enterprise Services se basan en la especificación original del modelo de datos de la empresa. Si bien hay cambios significativos, en el modelo de datos subyacente, se mantiene la esencia y las estructuras permanecen.

Par este trabajo se propone esta especificación para utilizarla en el sistema de gestión académico para realizar los procesos de matrícula.

1.3.1.3.3. IMS Learner Information Package (IMS LIP)

IMS Learner Information Package (IMS LIP) es una especificación de IMS que permite intercambiar información sobre un estudiante. Esta información es intercambiada entre diferentes sistemas, por otro lado se facilita la interoperabilidad entre estos sistemas y otros que soportan el ambiente de aprendizaje en Internet. Permite definir un conjunto de información del estudiante o de un productor de contenido de aprendizaje (autores, proveedores). La intención de esta especificación es definir un conjunto de paquetes que pueden ser usados para importar y extraer datos de estudiantes de un servidor compatible con IMS para lo cual define una estructura en XML para el intercambio de información sobre los alumnos

(individuales o grupos) entre diferentes LMS, sistemas de recursos humanos, sistemas de gestión del conocimiento, etc.

En esta especificación se indica qué información se almacena referente a un alumno (o grupo de alumnos) o incluso a un productor de contenido educativo, y cómo debe almacenarse.

La existencia de formatos consensuados para la definición de expedientes de alumnos permite su exportación entre sistemas educativos heterogéneos. Es necesario decidir qué información debe incluirse en el expediente y el formato para representarla. Dentro de los estándares para perfiles y expedientes debe contemplarse tanto la información estática (no depende de la interacción con el sistema, datos personales) como la dinámica (aquella que se genera o se modifica a medida que el alumno avanza en su proceso de aprendizaje, ej. calificaciones).

En la Figura 3 se puede observar la estructura de datos básica de un paquete IMS LIP. En los que se debe especificar los siguientes aspectos [33]:

- **Identificación:** En este se coloca la biografía y los datos demográficos relevantes para el aprendizaje.
- **Objetivo:** Sobre el aprendizaje, la carrera y otros objetivos y aspiraciones.
- **Títulos, Certificaciones y Licencias (qcl):** En este se sitúan las calificaciones, certificaciones y licencias concedidos por autoridades reconocidas.
- **Actividad:** Cualquier actividad relacionada con el aprendizaje en cualquier estado de finalización. Podría ser auto-reporte. Incluye la educación formal e informal, formación, experiencia laboral, y el servicio civil o militar.
- **Transcripción:** Un registro que se utiliza para proporcionar un nivel institucional basado en un resumen de los logros académicos. La estructura de este registro puede tomar de muchas formas.
- **Interés:** Información que describe los pasatiempos y actividades recreativas.
- **Competencia:** Habilidades, conocimientos y habilidades adquiridos en el desarrollo cognitivo, afectivo y / o dominios psicomotor.
- **Afiliación:** Miembro de las organizaciones profesionales, etc.

- **Accesibilidad:** Accesibilidad general de la información de los estudiantes, tal como se define a través de las capacidades del lenguaje, discapacidad y el aprendizaje cognitivo incluyendo preferencias (por ejemplo, cuestiones de estilo de aprendizaje), las preferencias de física (por ejemplo, una gran preferencia por la impresión), y preferencias tecnológicas (por ejemplo, una preferencia para una plataforma de computadora).
- **Llave de seguridad:** El conjunto de contraseñas y claves de seguridad asignadas a los alumnos para las transacciones con los alumnos, los sistemas de información y servicios.
- **Relación:** El conjunto de las relaciones entre los componentes básicos. El núcleo no dispone de estructuras dentro de los mismos identificadores que enlazan con las principales estructuras. En lugar de todas esas relaciones se plasman en un único núcleo de la estructura de modo que los vínculos más simples de identificar y gestionar.

Información del aprendiz/principiante	
Tipo de contenido ?	
Identificación *	Competencia *
Objetivo *	Accesibilidad *
TCL *	Transcripción *
Actividad *	Afiliación *
Interés *	Llave de seguridad *
Relación *	

Figura 3. Estructura de datos básica de un paquete IMS LIP.

Estas categorías han sido elegidas para cumplir los requisitos de una gran variedad de casos de uso y para facilitar la cartografía entre IMS y otras especificaciones pertinentes. Dentro de cada categoría los datos y la estructura de los elementos están definidos. Algunos de estos se especifican explícitamente como tipos de datos (cadenas de idioma, en su mayor parte) y otros se definen como estructuras jerárquicas recursivas. Además, los datos pueden ser definidos por los mecanismos de referencia. La referencia a los mecanismos de apoyo son referencias internas, las referencias a un sistema de información alumno externo, y las referencias a través de un URI.

Esta especificación se propone para la solución en el intercambio de información de los estudiantes o involucrados en el aprendizaje, principalmente entre el gestor académico y el LMS.

1.3.1.3.4. IMS Question & Test Interoperability

Esta especificación describe la forma de representar preguntas individuales o ítems (assessment item) y gestionar evaluaciones o exámenes completos (assessment). Su objetivo es conseguir que tanto las evaluaciones como los resultados sean intercambiables entre los diferentes LMS. Así, se podría disponer de almacenes de preguntas y bases de datos con los resultados obtenidos por los alumnos a los que cualquier sistema de enseñanza electrónica que se habilite podrá acceder.

IMS QTI define una estructura de datos XML para almacenar las preguntas independientemente del sistema o la herramienta de autoría utilizada para crearlas. Esto permite, por ejemplo, la integración de las preguntas o exámenes desarrollados con distintas herramientas en un único LMS. Por otro lado se propone un sistema coherente para que los sistemas puedan informar de cuál es el resultado de una evaluación.

Esta especificación permite la construcción de almacenes o repositorios de preguntas que sean directamente utilizables en distintos sistemas (e incluso para crear exámenes tipo test que los alumnos realicen por escrito).

1.3.1.4. ANSI / NISO Z39.50

ANSI / NISO Z39.50 es un protocolo para la búsqueda y recuperación de información basado en la estructura cliente/servidor que facilita la interconexión de sistemas informáticos, preferentemente es más conocido por Z39.50 por haber sido desarrollado por el comité número 39 de la American National Standards Institute (ANSI) y por ser el estándar 50 de la National Information Standards Organization (NISO).

Z39.50 tiene como objetivo fundamental permitir al usuario realizar búsquedas en bases de datos que cuenten con un servidor Z39.50, sin tener que conocer para ello, la sintaxis de búsqueda que utilicen dichos sistemas. Este estándar es utilizado para bibliotecas digitales y según [14], ofrece a los usuarios una interfaz de acceso única, que les permite tanto la búsqueda en el sistema local como en cualquier sistema remoto que soporte el mismo.

El estándar Z39.50, para desarrollar aplicaciones distribuidas, sigue las líneas del modelo cliente/servidor en el cual la parte cliente se llama “origen” y la parte servidora “destino”. Como se muestra en la Figura 5, si un usuario hace una petición en su sistema local, con la interfaz que éste le ofrece, el módulo origen Z39.50 que reside en su sistema traduce esta petición a un formato estandarizado y la emite a los sistemas remotos que convenga, siempre que disponga el mismo de un servidor Z30.50. Este servidor realiza la búsqueda en el sistema remoto, traduciendo de nuevo la petición al lenguaje de comandos o procedimientos que use el servidor, y devuelve el resultado al módulo origen [4]:

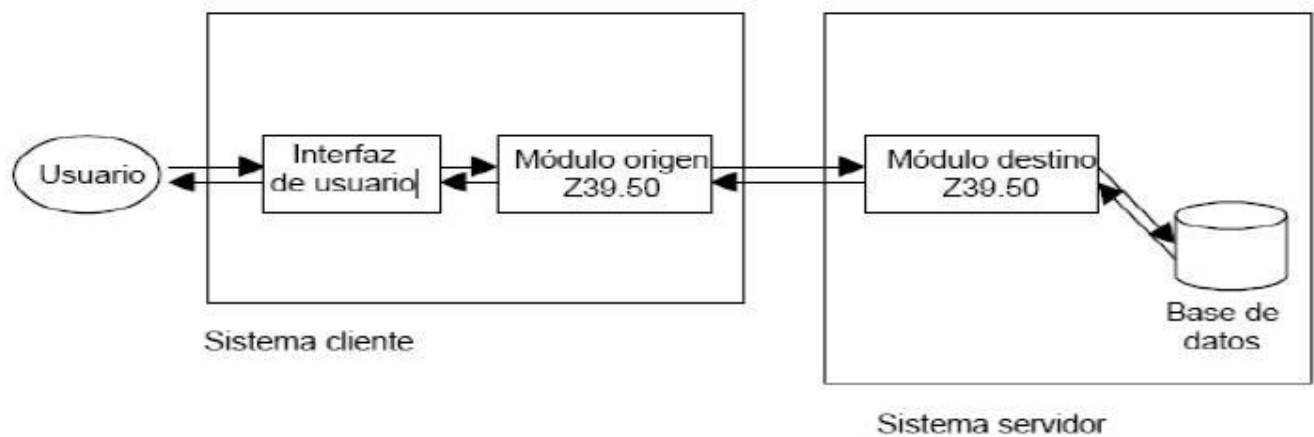


Figura 4. Diagrama de bloques de un sistema Z39.50.

Para lograr el intercambio de mensajes entre cliente y servidor, se realizan 11 servicios que permiten la transferencia de información, estos son: initialization, search, retrieval, result-set-delete, access control, accounting/resource control, sort, browse, explain, extended services, termination [4].

Z39.50 tienen una funcionalidad más completa que la OAI-PMH (será explicada en el próximo acápite), por ejemplo, tratan cuestiones como el manejo de sesiones, gestión de conjuntos de resultados y permiten la especificación de predicados para filtrar los resultados obtenidos. Sin embargo, esta funcionalidad acarrea un incremento en la complejidad de la implementación, genera un flujo de datos por las redes, produciendo demoras y problemas de tráfico así como un aumento en los costes. Independientemente de las desventajas que acarrea este estándar en su implementación, el mismo será propuesto en la solución final con vistas a lograr la interoperabilidad y la comunicación de la BV con otras herramientas.

1.3.1.5. The Open Archives Initiative Protocol for Metadata Harvesting (OAI-PMH)

La Open Archives Initiative Protocol for Metadata Harvesting (denominado OAI-PMH) proporciona una interoperabilidad independiente de aplicaciones basado en la recopilación de metadatos. Hay dos clases de participantes en el marco de OAI-PMH: los datos de Proveedores de administración de sistemas que apoyen el OAI-PMH como medio de exposición de los metadatos y los proveedores de servicios de uso de metadatos recolectados a través de la OAI-PMH como base para la creación de servicios de valor añadido [34].

OAI-PMH es una interfaz sumamente sencilla para acceder a la información bibliográfica disponible en un archivo o repositorio. Por lo que cualquiera puede realizar una implementación del mismo para poner a disposición de la comunidad Internet los datos que hasta ese momento estaban ocultos en bases de datos o catálogos.

Las peticiones OAI-PMH se expresan como las peticiones HTTP, para emitir preguntas y conseguir respuestas entre un servidor o archivo y un cliente o servicio recolector de metadatos. Las solicitudes OAI-PMH deben presentarse utilizando los métodos GET o POST de HTTP para no imponer limitaciones en cuanto a la duración de los argumentos, existiendo una sola base de URL para todas las solicitudes, en la cual se especifica la dirección de URL de Internet y el puerto [4].

Existen seis peticiones que un cliente puede realizar a un servidor las cuales se muestran en la siguiente tabla [4]:

Peticiones	Descripción
getrecord	Utilizado para recuperar un registro concreto. Necesita dos argumentos: identificador del registro pedido y especificación del formato bibliográfico en que se debe devolver.
Identify	Utilizado para recuperar información sobre el servidor: nombre, versión del protocolo que utiliza, dirección del administrador, etc.
listIdentifiers	Recupera los encabezamientos de los registros, en lugar de los registros completos. Permite argumentos como el rango de fechas entre los que queremos recuperar los datos.
listRecords	Recupera los registros completos.
listSets	Recupera un conjunto de registros. Estos conjuntos son creados opcionalmente por el servidor para facilitar una recuperación selectiva de los registros. Sería una clasificación de los contenidos según diferentes entradas. Un cliente puede pedir que se recuperen solo los registros pertenecientes a una determinada clase. Los conjuntos pueden ser simples listas o estructuras jerárquicas
listMetadataFormats	Devuelve la lista de formatos bibliográficos que utiliza el servidor.

Tabla 2. Peticiones de un cliente que se pueden realizar en un servidor

Las cinco primeras peticiones (getrecord, Identify, listIdentifiers, listRecords y listSets) son creados opcionalmente por el servidor para facilitar una recuperación selectiva de los registros.

El protocolo OAI-PMH soporta múltiples formatos para expresar los metadatos, no obstante requiere que todos los servidores ofrezcan los registros utilizando Dublin Core no calificado, codificado en XML.

Este protocolo no trata aspectos relacionados con la gestión o autorización para el acceso de los clientes. El servidor deberá recurrir a métodos externos si desea limitar los clientes a los que sirva información y relacionado con esto, tampoco trata aspectos relacionados a cómo los clientes pueden localizar aquellos servidores que contengan los datos que necesitan.

1.3.1.6. Simple Query Interface (SQI)

Simple Query Interface⁸² (comúnmente llamado SQI) se trata de una especificación, enmarcada en el entorno de los ROAs, que define una capa para facilitar las búsquedas de contenidos.

Especifica un estándar y un Application Program Interfaces (API⁸³) para resolver las problemáticas de las búsquedas de contenidos digitales en entornos heterogéneos como Internet, el establecimiento de sesión y la realización de consultas síncronas y asíncronas, que define los servicios que un repositorio puede tener disponibles para recibir y responder consultas de otros repositorios.

Como en SQI, es de mayor interés hacer accesibles los repositorios vía web, lo más idóneo es proveer este servicio mediante los Servicios Web y SOAP los cuales serán explicados más adelante.

El API define una serie de primitivas que permiten a los usuarios, considerar los repositorios digitales como “peers” (pares) de una comunicación en la que un repositorio puede actuar como fuente de consultas y otro como objetivo de las mismas [35].

SQI no establece la forma en la que los repositorios realizan la catalogación ni almacenan los metadatos de sus contenidos, ni define la manera en la que resuelven las consultas. Pero detalla, en el API, el mecanismo necesario para configurar tanto el lenguaje utilizado en la búsqueda, como el lenguaje de la respuesta, es decir, de lo único de lo que se preocupa es de enviar y recibir consultas, no de la estructura de las mismas, lo cual significa que SQI es mínimo y sencillo de implementar en una gran variedad de sistemas.

Este estándar define tres servicios básicos como son: gestión de la sesión, servicios de autenticación y servicios de aplicación como gestión de las consultas y un conjunto de métodos que se encuentran agrupados por categorías las que se muestran en la siguiente Tabla [36]:

⁸² En español Interfaz de Consulta Simple.

⁸³ En español Interfaz de Programa de Aplicación.

Categoría	Descripción	Métodos
The query management	Métodos que permiten la configuración de los parámetros de consultas	setQueryLanguage(), setResultsFormat(), setMaxQueryResults(), setMaxDuration().
In a synchronous query	Métodos que retornan los resultados de las consultas, que permiten la elección de los números de resultados retornados por una consulta o permiten conocer el número total de resultados de una consulta.	synchronousQuery(), setResultsSetSize() getTotalResultsCount()
In an asynchronous query	En estos métodos el destino envían los resultados de consultas hacia la fuente y la fuente ha de indicar la localización antes de enviar una consulta asíncrona.	queryResultsListener() set-SourceLocation() Asynchronous Query()

Tabla 3. Método por categorías de SQL

Teniendo en cuenta las ventajas y características anteriormente expuestas sobre este estándar, en este trabajo la comunicación con el repositorio, estará basada en SQL tal y como fuese propuesto en el FI que rige esta investigación. Igualmente es utilizado por el FI el estándar OKI el cual se describe en la próxima sección.

1.3.1.7. Open Knowledge Initiative (OKI)

Open Knowledge Initiative (OKI), es el nombre que recibe el estándar surgido como iniciativa del MIT, otras instituciones de estandarización y varias universidades con el propósito de desarrollar y promover las especificaciones que describen cómo los componentes de un entorno de software son capaces de comunicarse unos con otros.

OKI detalla cómo utilizar los servicios y con qué restricciones, siendo una propuesta concreta para el desarrollo de los mismos, describiendo así muchos de los servicios que están presentes en la mayoría de entornos de aprendizaje.

OKI brinda una gran variedad de interfaces para la definición de servicios denominados OSID's (Open Service Interface Definitions) y facilita una implementación de referencia para cada uno de ellos.

1.3.1.7.1. Arquitectura Open Service Interface Definitions (OSID) de OKI

OSID (Open Service Interface Definition) son cada una de las interfaces de servicios desarrolladas por OKI, es decir, son especificaciones de carácter formal que definen cómo construir un sistema de e-learning y cuáles son los servicios que debería tener. Cada interfaz detalla cuál es la funcionalidad que cada servicio ofrece.

Las OSID en cuanto al lenguaje de implementación son neutras, por lo que se pueden cambiar a otros lenguajes o incluso sería factible implantarlas en forma de servicio web, no hacen exaltaciones acerca de los detalles de la implantación.

Estas guías OSID contiene un conjunto de especificaciones de acuerdo al servicio que se pretenda brindar, pero no es necesario tener implementado todas las interfaces, igualmente no hay obligación de utilizar o implementar todos los métodos que propone cada interfaz, lo que posibilita que se pueda aplicar una metodología de desarrollo incremental. Algunas de las guías OSID son las siguientes [37]:

- **osid_authentication:** Soporta invocar procesos de autenticación. La implementación de estos servicios son responsables de garantizar cualquier información apropiada para llevar a cabo la autenticación. Los servicios que maneja esta OSID permiten probar si un usuario es autenticado, para ello devuelve el identificador que corresponde a cada usuario, los tipos de autenticación posible y además destruyen todas las autenticaciones o solo aquellas de un tipo dado.
- **osid_authorization:** Esta OSID provee una forma de definir quién, está haciendo qué y cuándo.
- **osid_Repository:** Esta osid, abarca el almacenamiento y recuperación de contenido digital, denominado Asset⁸⁴ (que pueden ser: documentos, material del curso, el tema de evaluación, imágenes, vídeo de audio, etc.), así como información sobre los Asset.

⁸⁴ En español es *Activo*

- **osid_Identifier (Id):** Este servicio es realmente un servicio compartido por otros servicios. Se usa para crear, recuperar y guardar identificadores únicos. Hay muchos contextos que necesitan a un identificador sin ambigüedades, en este caso servirá para identificar estudiantes, cursos, títulos, etc.
- **osid_Agent:** Representa a un usuario o proceso que invoca un servicio. Un Agente es un representante. Representa a un usuario o a un proceso dentro del Sistema. Por ejemplo un Agente específico sería el estudiante David Poll. Éste tendrá un identificador (por ejemplo 1239), un nombre a visualizar (Poll, David) y un tipo que podría indicar, por ejemplo, que es un estudiante.
- **osid_Locale:** Este servicio, permite realizar las conversiones de cadenas de texto, unidades de medida, moneda, fechas, etc. a los de un país, región o área geográfica específica y en el idioma que corresponda.
- **osid_Configuration:** Este da soporte a la hora de configurar, administrar y visualizar los valores de configuración que tendrán las aplicaciones y/o los diferentes servicios.
- **osid_Logging:** En este servicio se utiliza cuando las aplicaciones necesitan de lectura y grabación de ciertos tipos de trazas para poder hacer una serie de controles del Sistema, así como del desarrollo, del rendimiento, para obtener datos de seguimiento, etc. Un mecanismo de este tipo tiene que permitir diferentes tipos de formatos de trazas, así como de diferentes tipos de prioridad de éstas.
- **osid_Messaging:** Servicio que permite el envío de mensajes para comunicar entre ellos los diferentes componentes o aplicaciones empresariales.
- **osid_transaction:** Proporciona los medios en el que múltiples sesiones OSID puedan coordinarse.
- **osid_cataloging:** Proporciona un medio para la organización de objetos en los catálogos, establece la definición básica de las funciones usadas en todas las OSIDs y ayuda en la capacidad de adaptación de las estructuras de catalogación.
- **Osid.hierarchy:** Proporciona un medio de acceso, gestiona las relaciones padre-hijo entre elementos y apoya que sean recursivo y con múltiples nodos de los padres. Las autorizaciones de los usuarios normalmente se almacena como una jerarquía.

Todas estas especificaciones son basadas en una funcionalidad específica, por lo que para este trabajo se propone utilizar las OSIDs: `osid_authorization`, `osid_authentication`, y `osid_Logging` las que serán analizadas en el próximo capítulo. Parte considerable de este trabajo se basa en el FI, el cual propone en su solución la utilización de las dos primeras OSIDs de OKI mencionadas con anterioridad.

Los estándares vistos hasta el momento solo se encargan de facilitar la interacción e integración de herramientas e-learning y la interoperabilidad entre dichas herramientas de manera general. Aún cuando estos estándares permiten que varios sistemas se comuniquen, los mismos no garantizan que esta comunicación se lleve a cabo de forma segura. En los próximos acápite serán abordados algunos estándares de seguridad cuyo objetivo es proveer mecanismos para la comunicación e intercambio de información y datos entre aplicaciones de una manera segura.

1.3.2. Estándares de Seguridad

La seguridad es un requisito indispensable cuando se desea garantizar que la comunicación entre aplicaciones, los servicios web y el flujo de datos en general que fluye a través de la web se realice de manera segura y confiable, tanto como sea posible. Por este motivo el área de seguridad crece rápidamente, pero hay muchas dificultades a la hora de manejar datos con estructuras jerárquicas y con subgrupos de datos con diferentes requisitos en lo que se refiere a confidencialidad, derechos de acceso o integridad.

Es por ello que la seguridad debe contar al menos con los siguientes requisitos:

- **Confidencialidad:** garantizar que la información no es visible a extraños.
- **Autenticación:** garantiza que el acceso a la información sólo puede ser realizado por quienes proporcionan la identidad adecuada.
- **Integridad:** asegurar que el mensaje no ha sido modificado accidental o deliberadamente.
- **No repudiación:** garantiza que el emisor del mensaje no puede negar haberlo enviado.

A continuación se describen algunos de los principales estándares desarrollados para hacer frente los problemas antes mencionados.

1.3.2.1. Web Service Security (WS-Security, WSS)

El estándar Web Service Security o WSS fue publicado por la OASIS (Organization for the Advancement of Structured Information Standards) en el año 2004. WS–Security es un protocolo de comunicación que provee los medios necesarios para aplicar seguridad a los servicios web. Este protocolo contiene especificaciones que de alguna manera garantiza la integridad y seguridad en mensajería entre servicios web. WSS contiene detalles referentes al uso de SAML (Security Assertion Markup Language) y Kerberos, un protocolo de autenticación de redes de ordenador que permite a dos computadores en una red insegura demostrar su identidad mutuamente de manera segura, WSS incluye además formatos de certificado como X.509, que es un estándar UIT-T ⁸⁵ para infraestructuras de claves públicas, que especifica, entre otros aspectos, formatos estándar para certificados de claves públicas y un algoritmo de validación de la ruta de certificación [38]. Este estándar puede utilizar otras tecnologías de encriptación tal como la Public-Key Infrastructure-PKI, y el protocolo Secure Socket Layer / Transport Layer Security (SSL/TLS) que provee lo básico para dotar de seguridad extremo a extremo a las comunicaciones sobre la Internet.

WS–Security describe mejoras relacionadas con la seguridad aplicadas al SOAP messaging que atienden a la integridad y la confidencialidad del mensaje, incorporando para ello nuevas características en el encabezado de un mensaje SOAP y puede utilizar XML Signature⁸⁶ para definir la manera como se firmará el mensaje (las claves usadas y la firma), otra información puede ser cifrada con XML Encryption e incluirse en el encabezado de WS-Security. WSS transfiere las credenciales de los clientes a partir del *UserNameToken*, o cuando se utilizan datos binarios se usa *BinarySecurityToken* [38]. Integridad significa que un mensaje SOAP no es manipulado en su viaje desde un cliente hasta su destino final. Confidencialidad significa que un mensaje SOAP sólo es visualizado por los receptores a los que está destinado.

⁸⁵ El Sector de Normalización de las Telecomunicaciones de la Unión Internacional de Telecomunicaciones (UIT-T).

⁸⁶ Se verá más adelante

Este estándar presenta como inconvenientes el hecho de que solo se firma o codifica una parte del mensaje en vez del mensaje completo y que el mensaje puede ser enviado a través de diferentes protocolos de transporte HTTP, SMTP, FTP y TC, pero aún así garantiza la seguridad en los servicios web pues para la propuesta de este trabajo se tienen en cuenta este estándar.

1.3.2.2. XML Encryption

Uno de los formatos de mayor uso mundialmente para la distribución y el intercambio de información es sin duda el XML que a pesar de ser inicialmente un formato de documento, el mismo no contiene elementos vinculados a la seguridad.

Por todo ello surge XML Encryption como una recomendación del consorcio web (W3C) capaz de especificar un proceso de cifrado de datos y representar la información cifrada a su vez en un XML apto para viajar de manera segura por los medios de transmisión.

XML Encryption es un lenguaje cuya función principal es asegurar la confidencialidad de partes de documentos XML a través de la encriptación parcial o total del documento transportado. XML Encryption se puede aplicar a cualquier recurso Web, incluyendo contenido que no es XML [39].

XML Encryption establece que para cifrar un elemento (o un contenido textual de un elemento), este va a ser sustituido por otro elemento establecido por la especificación que contendrá el dato cifrado y la información necesaria para el procesamiento, como puede ser la identificación del algoritmo utilizado o datos sobre la clave empleada. Este elemento es *<EncryptedData>*, de tal modo que podría aparecer como raíz del nuevo documento generado tras el proceso, o como hijo de otros elementos ya existentes previamente [40]. Otros elementos contenidos dentro de *<EncryptedData>* son:

- EncryptionMethod.
- KeyInfo.
- CipherData.

Por otro lado, la especificación establece XML Schema (un lenguaje de esquema utilizado para describir la estructura y las restricciones de los contenidos de los documentos XML de una forma muy precisa, más allá de las normas sintácticas impuestas por el propio lenguaje XML.) como esquema de modelo de datos. Dicho Schema puede ser consultado en la especificación de XML Encryption y es necesario para conocer la sintaxis a emplear.

Así mismo XML Encryption permite granular el cifrado de documentos XML. Establece que es posible cifrar datos a distintos niveles de granularidad, desde elementos simples hasta documentos enteros. Algunos niveles de cifrado para un documento tienen que ver con:

- Cifrando de un elemento: Para cifrar un único elemento.
- Cifrando elementos hijos: Se puede optar por no cifrar todo el elemento completo sino algunos datos en concreto.
- Cifrando contenido textual: Si se desea proteger por ejemplo el contenido textual de un único elemento en particular.
- Cifrando documentos enteros : Cuando se necesita que toda la información sea protegida es posible cifrar todo un documento XML
- Súper-cifrado (cifrado datos previamente cifrados): Dado que es posible cifrar cualquier tipo de información, nada nos impide cifrar datos que ya estaban cifrados. Esto se conoce como Súper-cifrado y puede ser utilizado siempre que se respete la siguiente limitación: se debe cifrar el elemento <EncryptedData> al completo.

Este estándar es de mucha importancia para el cifrado de documentos XML por lo que se propone para la solución dado que WS Security lo utiliza en el cifrado de los datos del mensaje SOAP.

1.3.2.3. XML Digital Signature

Desarrollado por la W3C; provee mecanismos y especificaciones para crear firmas digitales basadas en XML. Este estándar asegura la integridad de partes de documentos XML transportados y define un esquema para capturar el resultado de la operación de firmas digitales y aplicarlas a los metadatos. También proporciona la autenticación de mensajes y/o servicios de autenticación de firma para datos de

cualquier tipo, tanto si se encuentra en el XML que incluye la firma o en cualquier otra parte [38]. Puede aplicarse a cualquier contenido digital (objeto de datos), incluyendo XML.

Lo que hace principalmente XML Signature es asociar claves con los datos de consulta. XML Signature representa un sistema que a través de una firma digital permite ofrecer autenticidad de los datos. Con la firma digital se confirma la identidad del emisor, la autenticidad del mensaje y su integridad, sin olvidar que los mensajes no serán repudiados.

XML Digital Signature permite demostrar la identidad y autenticidad del documento, tiene por objetivo: lograr la integridad y autenticación. Este estándar se rige por la siguiente estructura [39]:

- SignedInfo contiene la información que es firmada. Incluye:
 - CanonizationMethod:** algoritmo utilizado para canonizar el elemento
 - SignatureMethod:** algoritmo para convertir signedInfo en signatureValue
 - Reference:** incluye el método para obtener la firma (digest) y el valor, así como otras transformaciones realizadas.
- SignatureValue incluye el valor de la firma.
- KeyInfo indica la clave a utilizar para validar la firma.

Este estándar posee disímiles ventajas que se han expuesto anteriormente por lo que se propone utilizarlo en la solución para el firmado de los mensajes SOAP y sin olvidar la desventaja que presenta este, de no contener la certeza sobre la fecha y la hora en la que el documento fue firmado.

1.3.2.4. Extensible Rights Markup Language (XrML)

La Organización para el Avance de Estándares de Información Estructurados (OASIS) facilitó la disponibilidad del Extensible Rights Markup Language (XrML) como un estándar para la especificación y gestión de derechos, unido a fuentes tales como contenidos digitales y servicios web.

XrML (eXtensible Rights Markup Language), puede ser utilizado en contenidos centrales, así como en los modelos de negocio basados en los servicios. Estandarizado por International Standards Organization (ISO) como MPEG-21 Rights Expression Language, está siendo revisado por la OASIS y ya cuenta con el soporte de firmas como Xerox, Microsoft, Adobe, Audible.com, Hewlett-Packard, OverDrive y Portal Software en programas de tratamiento de textos, edición, gestión del contenido y otros productos de seguridad [41].

Como principal ventaja, cabe destacar que el estándar soporta interoperabilidad entre plataformas DRM (Digital Rights Management) mientras permite establecer modelos de acceso que se extienden desde lo simple a lo complejo.

XrML permite usar políticas sobre contenidos digitales en la forma de licencias. Estas licencias XrML definen quién puede acceder a un contenido determinado y cómo se protege y distribuye; asimismo, controlan los derechos de uso en múltiples detalles, como los permisos temporales y las operaciones para las que se autoriza la impresión [41]. Para ello, cuando un autor protege un contenido, que puede estar en múltiples formatos, desde los de documento de texto o de hoja de cálculo a informes Web, se encripta para prevenir accesos no autorizados. XrML, acelerará la distribución de contenido digital y las iniciativas de servicios web, permitiendo la interoperabilidad entre los componentes del sistema o servicios.

Una limitación de XrML es que carece de métodos para realizar seguimientos de las acciones de los usuarios sobre los contenidos protegidos, y en consecuencia, realizar auditorías. Por otro lado, esta norma aporta un mecanismo aceptable para la creación de una estructura común que permita definir y expresar los derechos que protegen el contenido electrónico.

Los siguientes pasos indican el funcionamiento de XrML, el cual expresa los derechos y condiciones que protegen el contenido electrónico de usos no autorizados [41]:

1. El servidor de licencias lanza al servidor de contenidos las licencias de acceso y uso sobre un contenido específico.
2. El servidor de licencias ofrece licencias a diferentes aplicaciones que controlan el uso de los contenidos por parte de cada una de ellas.

3. El servidor de credenciales proporciona licencias de credenciales a aquellos usuarios que comunican su identidad o el rol que desempeñan.

4. Las aplicaciones A y B pueden tener acceso al mismo contenido protegido. El uso de un lenguaje único de derechos permite la interoperabilidad.

1.3.2.5. Security Assertion Markup Language (SAML)

Con el advenimiento de tecnologías como SOAP, XML-RPC y "Servicios Web", SAML proporciona un idioma común para el despliegue de toda la infraestructura de seguridad.

SAML (Security Assertion Markup Language) permite compartir información de autenticación entre aplicaciones y se encuentra basado en autoridades que emiten aserciones. Transmite varios tipos de aserciones [42]:

- Aserción de autenticación
- Aserción de atributo

Es un framework que permite intercambiar información de identificación, como infraestructuras de llave pública PKI. Es un entorno basado en XML que facilita el intercambio de autenticación y autorización [38]. Esencialmente, el estándar formula un conjunto de procesos para intercambiar los datos de identificación de los usuarios y de los controles de acceso. Ha sido propuesto por fabricantes de productos de seguridad (PKI, Single Sign-On) como Tivoli, Entegrity, RSA, EnTrust, etc.

SAML proporciona en esencia un protocolo de mensajes tipo pregunta-respuesta para que los dominios de seguridad implicados gestionen las decisiones necesarias de autenticación y autorización. Aplicado al uso de servicios web, las aserciones de SAML son válidas en los mensajes SOAP para proporcionar seguridad en las transacciones de web services, soportando firma única (SSO o Single Sign-On). Single Sign-On es un procedimiento de autenticación que habilita al usuario para acceder a varios sistemas con una sola instancia de identificación.

De entre las ventajas que incorpora este framework, es muy importante destacar el nivel de abstracción que supone en el modelo de seguridad y que le hace independiente de plataformas y soluciones comerciales particulares por lo que se propone este estándar para que se utilice conjuntamente con WS-Security para intercambiar información de autenticación entre aplicaciones.

1.3.2.6. XML Key Management (XKMS)

XML Key Management (Especificación XML para manejo de claves, XKMS en idioma inglés) es un protocolo para distribuir y registrar claves públicas [38]. Lo que hace es ocultar la parte compleja que surge con PKI (Infraestructura de Clave Pública). Además es compatible para su uso en conjunto con XML-SIG (Especificación XML de firma digital) y XML-Enc (Especificación XML de cifrado). XKMS tiene por objetivo principal: ocultar al usuario la complejidad subyacente en PKI.

XKMS consta de dos partes o dicho de otra manera, la especificación XKMS encapsula 2 estándares las que se muestran a continuación [38]:

- **Registro de clave pública**

XKISS - XML Key Information Service Specification (Especificación XML de Servicio de Información de claves). Este estándar define un conjunto de protocolos para el procesamiento, y despacho de la información asociada a las llaves en un firmado XML (XML Signature) y la información de las claves públicas y privadas de los datos cifrados (XMLEncrypted). Procesar el campo KeyInfo contenido en las firmas digitales.

- **Información de clave pública**

XKRSS - XML Key Registration Service Specification (Especificación XML de Servicio de registro de claves). Especifica un conjunto de protocolos que soportan el registro de un par de claves para las gestiones de infraestructura pública y privada. El Objetivo de X-KRSS es responder a los requerimientos completos para gestionar todo el ciclo para la gestión del intercambio de claves, en general, estos pasos

pueden ser registro, revocación y recuperación de claves. La generación de claves se puede realizar por el cliente o al momento del registro en el servidor. Buscar la clave pública de alguien y comprobar si es buena.

1.3.2.7. Extensible Access Control Markup Language (XACML)

Extensible Access Control Markup Language (XACML) fue aprobado en febrero de 2003, XACML está diseñado para que las políticas de control de accesos no estén escritas en lenguajes propietarios específicos de cada dispositivo o aplicación. Incluye un lenguaje de control de accesos y un lenguaje de pregunta/respuesta que permite a los desarrolladores escribir políticas, que dicten los usuarios que pueden acceder a los recursos internos o de Internet. Para ello, la norma estará presente en cortafuegos, servidores y software de gestión de accesos Web, así como en gateways o pasarelas, que conecten dispositivos de políticas de control de accesos. XACML se complementa con SAML para ofrecer autenticación, gestión de identidades y autorización de servicios Web.

En este contexto los estándares SAML y XACML estandarizan el intercambio de los datos de autenticación y autorización generados por el proveedor de identidad e interpretados por las aplicaciones.

1.4. Arquitectura Orientada a Servicio

Evolucionada de la computación distribuida, una Arquitectura Orientada a Servicios (SOA) se encuentra basada en el paradigma de pregunta / respuesta para aplicaciones síncronas y asíncronas. A su vez, este tipo de arquitectura brinda una forma estándar de exposición e invocación de servicios (no exclusivamente servicios web), lo cual facilita la interacción entre diferentes sistemas propios o de terceros, permitiendo que las funciones o lógica de negocio de las aplicaciones sean presentadas como servicios a otras aplicaciones consumidoras /clientes. Su aporte respecto a modelos anteriores es el principio fundamental que rige en su práctica: Agilidad del negocio.

1.4.1. Características de SOA

Una de las principales características presentes en los servicios que ofrece una arquitectura SOA tiene que ver con su naturaleza desacoplada (o bajo acoplamiento), lo cual significa que la interfaz de dichos servicios es indiferente de la implementación de los mismos, es decir, un servicio puede ser implementado en un determinado lenguaje y la aplicación que consume este servicio puede estar implementada o soportada por un lenguaje o plataforma totalmente diferente a la de este servicio, permitiéndole así un alto grado de interoperabilidad [43]. En otras palabras, SOA facilita la comunicación y el intercambio de datos, entre medios (aplicaciones) heterogéneos al abstraerse completamente de plataformas, hardware, sistemas operativos y lenguajes de programación, facilitando la integración de disímiles tecnologías. Con esta arquitectura, se pretende que los componentes software desarrollados sean muy reusables, ya que la interfaz se define siguiendo un estándar.

Toda arquitectura SOA tiene las siguientes características [44]:

- Los servicios SOA auto describen su interfaz en documentos XML que son independientes de la plataforma. WSDL (Web Service Description Language) es el lenguaje utilizado para describir los servicios.
- Los servicios SOA se comunican con mensajes formalmente definidos vía un esquema XML. La comunicación entre consumidores y proveedores de servicios típicamente ocurre en un ambiente heterogéneo, con poco o ningún conocimiento sobre el proveedor.
- Los servicios SOA son mantenidos en una empresa por un registro que actúa como un directorio. Las aplicaciones pueden mirar los servicios en este registro e invocarlos. UDDI (Universal Description, Definition and Integration).
- Cada servicio SOA tiene una calidad de servicio asociada con el (QoS). Algunos de los ejemplos clave de la QoS son los requerimientos de seguridad, como la autenticación y la autorización, mensajería confiable y políticas definiendo quién puede acceder a los servicios.

Otros tipos de arquitecturas de software, con el objetivo de evitar en lo posible la modificación de las mismas por los costes que esto implicaría, se ven forzadas a implementar middleware o pasarelas que le

permitan acceder a los servicios de otras aplicaciones. Esta, a pesar de ser una solución, no es la idónea para comunicar aplicaciones heterogéneas que no cuentan con una arquitectura SOA.

Dos de los principales elementos de una arquitectura SOA refieren a las funciones y a la calidad de servicios. Dentro de las funciones [45] se identifican:

- **Transporte:** es el mecanismo utilizado para llevar las demandas de servicio desde un consumidor de servicio hacia un proveedor de servicio, y las respuestas desde el proveedor hacia el consumidor.
- **Protocolo de comunicación de servicios:** es un mecanismo acordado a través del cual un proveedor de servicios y un consumidor de servicios comunican qué está siendo solicitado y qué está siendo respondido.
- **Descripción de servicio:** es un esquema acordado para describir qué es el servicio, cómo debe invocarse, y qué datos requiere el servicio para invocarse con éxito.
- **Servicios:** describe un servicio actual que está disponible para utilizar.
- **Procesos de Negocio:** es una colección de servicios, invocados en una secuencia particular con un conjunto particular de reglas, para satisfacer un requerimiento de negocio.
- **Registro de Servicios:** es un repositorio de descripciones de servicios y datos que pueden utilizar proveedores de servicios para publicar sus servicios, así como consumidores de servicios para descubrir o hallar servicios disponibles.

Y dentro de Calidad de Servicio (QoS) quedan recogidos los elementos:

- **Política:** es un conjunto de condiciones o reglas bajo las cuales un proveedor de servicio hace el servicio disponible para consumidores.
- **Seguridad:** es un conjunto de reglas que pueden aplicarse para la identificación, autorización y control de acceso a consumidores de servicios.
- **Transacciones:** es el conjunto de atributos que podrían aplicarse a un grupo de servicios para entregar un resultado consistente.
- **Administración:** es el conjunto de atributos que podrían aplicarse para manejar los servicios proporcionados o consumidos.

Otros atributos de QoS como la coordinación entre socios y las transacciones que involucran múltiples servicios están siendo tratados en las especificaciones WS-Coordination y WS-Transaction.

Un Bus de Servicios Empresariales (ESB, Enterprise Service Bus) es uno de los componentes presentes en una arquitectura SOA. Dicho componente proporciona una comunicación fiable entre los distintos recursos tecnológicos tales como aplicaciones, plataformas y servicios, que están distribuidos en múltiples sistemas por toda la empresa. Los ESB constituyen los cimientos de una SOA y pueden complementarse con capacidades de productividad adicionales, como la orquestación de servicios y los registros. Es además un elemento de integración que combina mensajería, servicios web, transformación de datos, enrutamiento, políticas de seguridad, entre otras. [46] Identifica las principales características de un ESB:

- Validación de la información, de los mensajes entrantes, sobre los que se pueden aplicar distintas condiciones y verificaciones que hagan cumplir los requisitos establecidos.
- Enriquecimiento de los mensajes, posibilidad de añadir información o meta información adicional para cumplir las necesidades de integración en cada momento.
- Transformación de los mensajes, normalmente entre los diversos modelos de datos del conjunto de servicios y aplicaciones a integrar.
- Ruteo: la mayoría de los ESB permiten definir procesos, ofreciendo alguna (mínimas) lógica de flujos, como estructuras iterativas, condicionales, procesamiento en paralelo, invocación de otros servicios, entre otras.
- Seguridad Incorporada: normalmente cada ESB trae incorporada su propia seguridad referida a la autenticación y autorización de usuarios. También soportan protocolos de comunicación segura, mediante HTTPS.
- Interfaz de usuario: algunos ESB brindan la posibilidad de administrar el bus a través de una interfaz amigable.
- Orquestación de servicios: a través de lenguajes estándares como BPEL, permiten la coordinación y organización de la invocación de los distintos servicios. De esta manera es posible definir procesos del negocio dentro de los ESB.

Igualmente, los Business Process Management (BPM) son otro tipo de componentes. Con la aplicación de los **BPM**, las áreas orientan sus necesidades en torno a procesos de negocio, independientemente de la

tecnología. Los procesos de negocio se componen en unidades de trabajo bien definidas, con una función específica y deben poder aplicarse en el contexto de diferentes procesos, es por ello que los BPM se encargan de facilitar el análisis, definición, ejecución, monitoreo, y control de los mismos.

Los BPM permiten el ahorro de tiempo y costes de programación, los procesos se ejecutan, monitorizan y optimizan directamente por el personal de una empresa, y permite que los procesos puedan rehacerse indefinidamente sobre la marcha para adaptarse a nuevas necesidades o mejoras. [47].

1.4.2. Niveles de Madurez de SOA

Otro aspecto a tener en cuenta son los niveles de madurez de SOA, los que homológamente igualan en número y tienen relación directa con los niveles de madurez que propone CMM. Efectivamente, SOA cuenta con 5 niveles que a diferencia de CMM estos se focalizan más a aspectos técnicos referentes de la propia arquitectura. Estos niveles son:

- **Initial Services:** Indica fases de exploración y adopción SOA, se logra que en los proyectos de desarrollo se integre SOA como parte de la arquitectura, se utilizan como estándar WSDL, SOAP, XML, J2EE y .NET, en este nivel existen servicios creados para necesidades inmediatas NO para servicios de negocio. Se lleva a cabo mayormente en proyectos pequeños.
- **Architected Services:** En este nivel entra la mediación de servicios, aquí entra en juego las características de un ESB (producto que se obtiene en este nivel), mensajería confiable (WS-RM), transformación (Xquery), etc. Además de contar con un registro de los servicios como lo es UDDI. En este nivel se encuentra la integración a bases de datos mediante EII. Una meta a alcanzar en este nivel es reducir los costos y lograr que múltiples aplicaciones sean integradas sin importar la heterogeneidad de las mismas.
- **Business Services/Collaborative Services:** Para estar en este nivel, ya se debe contar con el desarrollo de servicios de negocio. Existe también la colaboración de servicios para lograr procesos que den mayor sentido a la organización así como aumentar la agilidad del negocio, aquí es donde tiene sentido lo que se conoce como Business Process Management (BPM). Utiliza como estándares a WS-BPEL.

- **Mesured Business Services:** Aquí el punto principal es el monitoreo de procesos. En este nivel puedes tener métricas de los procesos de negocio basados en servicios.
- **Optimized Business Services:** Dado que ya se obtienen métricas de los procesos de negocio en el nivel que le precede, este nivel tiene como propósito la optimización de estos procesos. Aquí entra una cultura de mejoramiento continuo.

1.4.3. ¿Qué son los Servicios Web?

Un Web Service (Servicio Web en español) es un conjunto de protocolos y estándares que sirven para intercambiar datos entre aplicaciones. Diversidad de aplicaciones y software hacen uso y se benefician de los servicios web dado la flexibilidad y ventajas que ofrecen los mismos. Los servicios web son una buena práctica de programación que permite intercambiar datos e información en una red de computadores (ejemplo internet), los mismos son capaces de ejecutarse sobre diferentes plataformas y facilita la comunicación entre aplicaciones implementadas en diversos lenguajes de programación.

Algunas de las ventajas que ofrecen los servicios Web son:

- Aportan interoperabilidad entre aplicaciones de software independientemente de sus propiedades o de las plataformas sobre las que se instalen.
- Los servicios Web fomentan los estándares y protocolos basados en texto, que hacen más fácil acceder a su contenido y entender su funcionamiento.
- Al apoyarse en HTTP, los servicios Web pueden aprovecharse de los sistemas de seguridad firewall sin necesidad de cambiar las reglas de filtrado.
- Permiten que servicios y software de diferentes compañías ubicadas en diferentes lugares geográficos puedan ser combinados fácilmente para proveer servicios integrados.
- Permiten la interoperabilidad entre plataformas de distintos fabricantes por medio de protocolos estándar y abiertos. Las especificaciones son gestionadas por una organización abierta, la W3C, por tanto no hay secretismos por intereses particulares de fabricantes concretos y se garantiza la plena interoperabilidad entre aplicaciones.

A continuación se describen brevemente algunos estándares empleados por los servicios web.

1.4.4. Estándares empleados por los servicios web

Organizaciones como OASIS y W3C son los comités responsables de la arquitectura y reglamentación de los servicios Web. Para mejorar la interoperabilidad entre distintas implementaciones de servicios Web se creó el organismo WS-I, encargado de desarrollar diversos perfiles para definir de manera más exhaustiva estos estándares.

- **Web Services Protocol Stack:** Denominación que recibe el conjunto de servicios y protocolos de los servicios Web.
- **XML (Extensible Markup Language o «lenguaje de marcas ampliable»):** Es el formato estándar para los datos que se vayan a intercambiar y permite la compatibilidad entre sistemas, facilitando la información de una manera segura, fiable y fácil.
- **SOAP (Simple Object Access Protocol) o XML-RPC (XML Remote Procedure Call):** Protocolos sobre los que se establece el intercambio. SOAP es un protocolo que establece un formato común de mensajes para el intercambio de datos entre clientes y servicios.
- **Otros protocolos:** Los datos en XML también pueden enviarse de una aplicación a otra mediante protocolos normales como HTTP (Hypertext Transfer Protocol), FTP (File Transfer Protocol), o SMTP (Simple Mail Transfer Protocol).
- **WSDL (Web Services Description Languages):** Es el lenguaje de la interfaz pública para los servicios Web. Es una descripción basada en XML de los requisitos funcionales necesarios para establecer una comunicación con los servicios Web listados en su catálogo. Las operaciones y mensajes que soporta se describen en abstracto y se ligan después al protocolo concreto de red y al formato del mensaje.
- **UDDI (Universal Description, Discovery and Integration):** UDDI es uno de los estándares básicos de los servicios Web cuyo objetivo es ser accedido por los mensajes SOAP y dar paso a documentos WSDL, en los que se describen los requisitos del protocolo y los formatos del mensaje solicitado para interactuar con los servicios Web del catálogo de registros. Permite comprobar qué servicios web están disponibles.
- **WS-Security (Web Service Security):** Protocolo de seguridad aceptado como un estándar por OASIS (Organization for the Advancement of Structured Information Standards). Garantiza la

autenticación de los actores y la confidencialidad de los mensajes enviados. Este estándar fue tratado en secciones anteriores de este acápite con mayor amplitud.

A continuación serán analizados algunos de los componentes de UVs existentes en la Universidad de las Ciencias Informáticas (UCI).

1.5. Componentes de UVs existentes en la UCI

No existe en la actualidad una Universidad Virtual para la UCI pero en ella se dispone de un conjunto de componentes, algunos en funcionamiento y otros en desarrollo así como novedosas líneas de investigación sobre las más diversas temáticas referentes a la educación a distancia y la virtualización de la enseñanza.

En esta sección serán tratados aquellos componentes existentes en la UCI y que figuran como posibles candidatos a ser parte estructural en la propuesta de UV para la misma.

1.5.1. Biblioteca Digital de la UCI

En la UCI existe la Biblioteca Digital, que proporciona servicios especializados de búsqueda en catálogos y en bases de datos internacionales, sobre disímiles temas de la especialidad y otros aspectos y productos de información de alto valor agregado, que apoyan las funciones principales de la universidad referentes al proceso docente (docencia, investigación, extensión universitaria y producción), mediante la selección adecuada de las fuentes de información y la utilización de las TICs.

Esta biblioteca se clasifica dentro de las bibliotecas automatizada pues no todos sus procesos han borrado las dificultades del tiempo y el espacio pues aún hay procesos a los que hay que dirigirse personalmente al local, para erradicar estos inconvenientes se está realizando la Biblioteca Electrónica de la UCI la cual no va a estar orientada a servicio, pero si van a publicar tres servicios básicos como el servicio de Adquisición a través del cual se publican todos los materiales nuevos de la Biblioteca, el servicio de Impresión para realizar todos los procesos de presentación de documentos , las copias, etc., al

departamento de Impresión sin tener que dirigirse al lugar y el servicio de pregunta a Expertos, donde el estudiante realiza preguntas y un conjunto de expertos la analizan y las responden, todo esto de forma on-line.

A raíz de una entrevista realizada al líder de proyecto de la nueva versión de la biblioteca de la UCI (aún en desarrollo) se evidenció que la misma cuenta con varios módulos como:

- **Módulo de Catalogación:** En este módulo se recogen todos los datos del catálogo, además de agregar registros, autoridades y publicar periódicos.
- **Módulo de Circulación:** En este se realizan las devoluciones de los libros, transferencias, etc.
- **Módulo de Adquisición:** Este se relaciona con los nuevos libros de la Biblioteca.
- **Módulo de Administración:** Todos los procesos de administración del sitio.

Todos estos módulos se conciben con el objetivo de automatizar los procesos de biblioteca. Por otro lado utiliza un conjunto de estándares, por ejemplo el MARC 21 que define un formato de datos para el resultado de consultas bibliográficas, ISO 27/09 y Z39.50 para la recuperación de información basada en la estructura cliente/servidor. Debe resaltarse que hasta la fecha actual la Biblioteca no es interoperable con ningunos de los componentes UCI candidatos a una UV.

1.5.2. Akademos

Durante el curso docente que comprende al período 2004/2005, la UCI pone en funcionamiento un Sistema de Gestión académica que recibe por nombre Akademos (nombre de su primera versión). Esta primera versión del sistema se centra en una aplicación Web que usa tecnologías ASP.NET y Servicios Web XML para facilitar la interoperabilidad y la comunicación con otros sistemas, cuenta con siete módulos principales (ver figura 5) que permiten satisfacer las exigencias de los clientes. Con vista a una mejor comprensión de este sistema algunos de estos módulos se abordan a continuación [48]:



Figura 5. Estructura de los módulos de Akademos [48].

Módulo Plan de Estudio

Este módulo permite la creación del plan de estudio de la carrera definiendo los niveles divididos en momentos que los estudiantes deben vencer una vez concluidas las asignaturas. También permite la definición de los diferentes perfiles y disciplinas en las que se agrupan las asignaturas.

Módulo Matrícula

Este módulo es el encargado de mantener el control de los estudiantes una vez que ingresan al centro así como la gestión de los movimientos que se le aplican a un estudiante durante su paso por la universidad. El módulo además es el encargado de definir la estructura del centro de estudios en la cual se agrupa un conjunto de estructuras organizadas jerárquicamente.

Módulo Expediente

Este módulo almacena de forma digital un conjunto de información referente al desempeño académico y a otros aspectos del comportamiento de un estudiante en su paso por la universidad.

Módulo Reportes

Este módulo cuenta con diferentes herramientas que permiten al usuario el diseño y publicación de nuevos reportes que involucren diferentes aspectos de los estudiantes. Los reportes pueden ser generados dinámicamente u obtenidos de manera estática por parte de los usuarios que interactúan con este módulo, permitiéndoles personalizarlos de acuerdo a sus necesidades.

Módulo Registro

Este módulo es el encargado de permitir a un profesor llevar el control de las asistencias y las evaluaciones aplicadas a sus estudiantes.

Módulo Profesor

Este módulo permite mantener el control de la plantilla de profesores y la carga docente relativa a los mismos que no es más que las asignaturas que imparte en cada uno de los grupos que atiende.

Módulo Estudiante

Este módulo permite al estudiante interactuar con el sistema con el objetivo de consultar el registro de sus evaluaciones, así como realizar búsquedas de las evaluaciones de otros estudiantes.

El módulo de seguridad (no aparece en la figura anterior) gestiona y controla el acceso de los usuarios al sistema, verificando los permisos que poseen los mismos. De igual manera registra las incidencias de las acciones de los usuarios en su interacción con la aplicación.

A raíz de una entrevista realizada a la líder del proyecto Akademos, sale a relucir que la nueva versión de Akademos no utiliza estándares de interoperabilidad a pesar de brindar información a aplicaciones externas como son el Entorno Virtual de Aprendizaje, el Sistema de Control de Acceso, y los Directorios de Personas y Telefónicos, igualmente no cumple con todas las necesidades de gestión para los estudiantes de la universidad.

Por tales motivos actualmente se trabaja en la nueva versión funcional de Akademos, la cual además de mantener las funciones de la versión anterior, incluirá otras funcionalidades relacionadas con la matriculación y la titulación tanto para pregrado como para posgrados.

Las versión del SGA que sucederá a la actual versión en funcionamiento, no aprovechará las ventajas que ofrece una Arquitectura Orientada a Servicios (SOA). Su lógica de negocio no estará orientada a una arquitectura SOA; pero algunos procesos dentro del negocio sí serán implementados auxiliándose de los

Servicios Web. Como desventaja debe resaltarse el hecho de que no se utilizarán ni implementarán estándares de interoperabilidad para la integración de este SGA con otras herramientas y sistemas externos.

1.5.3. Entorno Virtual de Aprendizaje (Moodle)

Un componente presente en la UCI y con grandes posibilidades de insertarse como unidad medular en la propuesta de UV para esta institución, recae indudablemente sobre el Entorno Virtual de Aprendizaje (EVA), construido a partir de una plataforma tecnológica de e-learning como es el LMS moodle, una de las plataformas de aprendizaje más difundidos y exitosas a nivel mundial, el EVA permite acceder a diferentes espacios como: foros, chat, diálogos, actividades, etc. y permite adquirir una mayor comunicación entre quienes cursen la carrera y el profesorado responsable, para que el proceso de enseñanza-aprendizaje sea logrado con la mayor eficacia posible.

Generalizando, un EVA ofrece un espacio virtual donde los miembros de una comunidad educativa interactúan con la finalidad de desarrollar un proceso formativo mediante la aplicación de las TIC [49].

Por otro lado el EVA ofrece beneficios como los que se listan a continuación:

- Es un entorno centrado en la comunicación y las actividades de enseñanza/aprendizaje, incluyendo herramientas para gestionar materiales.
- Contiene una amplia flexibilidad y usabilidad debido al diseño modular del entorno que posibilita que pueda dar soporte a cualquier tipo de modalidad educativa y a que presta mayor atención a la interfaz del usuario.
- Está desarrollada en PHP y MySQL para Linux y Windows.

El diseño modular de moodle le confiere gran flexibilidad, ya que permite que otros módulos⁸⁷ le sean añadidos en pos de extender la gama de funcionalidades, beneficios y utilidades que ofrece. Tal es el caso de C2SCORM (Course to Site o Curso a SCORM), módulo concebido dentro del Polo Teleformación

⁸⁷ Componentes auto-controlados que extienden las funcionalidades de una aplicación.

y capacitado para insertarse como una solución factible dentro del EVA (moodle, versión 1.8.x) con la finalidad de compartir, reutilizar y exportar los contenidos de Moodle, permitiendo reutilizarlos en cualquier plataforma compatible. Para ello, es capaz de exportar dichos cursos a paquetes SCORM [50].

Referente a la interoperabilidad con otros sistemas, moodle trae habilitado estándares como el IMS ES que le confieren la capacidad de comunicarse con SGA (Sistemas de Gestión Académica). El Polo Teleformación, en su investigación FI propone y describe un modelo de arquitectura en la que se tienen en cuenta estándares de interoperabilidad como: SQL y OKI, los cuales posibilitan la comunicación con otros sistemas como ROA (Repositorio de Objetos de Aprendizaje) y ROXS (Herramienta de Autor), otras herramientas igualmente concebidas dentro de este polo.

1.5.4.Repositorio de Objetos de Aprendizaje

ROA, nombre que recibe el Repositorio de Objetos de Aprendizaje de la UCI, tiene como objetivo fundamental almacenar y gestionar los OA, proporcionando la reutilización de los mismos y reforzando el proceso de capacitación existente en cualquier institución; es una aplicación Web modular y multiplataforma. Dispone de un lugar común, accesible a través de un navegador, se rige por el estándar SCORM 1.2, cuenta con una amplia gama de funcionalidades que se agrupan en los siguientes módulos:

- Módulo para gestionar los paquetes SCORM que serán almacenados en el repositorio.
- Módulo para organizar los contenidos una vez subidos al repositorio, a través de categorías.
- Módulo para propiciar la comunicación entre los diferentes usuarios del repositorio, de acuerdo a su rol.
- Módulo para permitir búsquedas en el repositorio, basándose en los metadatos del estándar LOM.
- API (Application Programming Interface) para garantizar la interoperabilidad entre el repositorio y el ADA de PDVSA, basada en el estándar SQL.
- Módulo para generar reportes estadísticos sobre los usuarios y paquetes del sistema.
- Módulo para gestionar las cuentas de los usuarios y los privilegios correspondientes.
- Módulo para configurar las variables globales que inciden en el funcionamiento del sistema.
- Módulo para generar reportes con información sobre las acciones que realizan los usuarios del sistema.

La versión actual de ROA es capaz de intercambiar información con otras herramientas como el EVA y ROXS, basándose en los estándares IMS-DRI y SQL, a raíz del FI el cual constituye piedra angular en la combinación y comunicación de estos componentes.

1.5.5.Herramientas de Autor Web (ROXS)

Desarrollada en la UCI y como resultado del Polo Teleformación se obtiene una primera versión de la Herramienta de Autor: ROXS. A diferencia de otras herramientas de autor existentes en el mundo, tales como Reload (programadas para escritorio), Lompad, eXeLearning, etc. ésta es una aplicación web. Además, ROXS es de carácter modular y multiplataforma, que permite la creación de contenidos (paquetes) de acuerdo a las especificaciones SCORM 1.2 (utiliza LOM para la descripción con metadatos) a partir de OA, posibilitando la reutilización de los contenidos didácticos en herramientas e-learning que soporten este estándar. ROXS, al ser una herramienta web ofrece un entorno flexible, de fácil administración y uso general, con una interfaz amigable y accesible al usuario. Los paquetes SCORM generados por esta herramienta pueden ser utilizados en la elaboración de cursos dentro de un LMS y en cualquier herramienta que de manera general acepte o manipule recursos que sigan las normas de empaquetado SCORM.

Como inconvenientes, cabe resaltar que ROXS no hace uso de estándares en la implementación de una vía que garantice que esta herramienta sea interoperable a nivel de sistema con otras aplicaciones, aunque para ello fueron propuestos algunos mecanismos y soluciones que así lo permiten y las mismas se encuentran recogidas dentro de un FI. Los aportes de este framework son tomados como base en otras investigaciones que también involucran a ROXS como herramienta de autor. Esta herramienta aún está en fase de desarrollo

1.6. Conclusiones del Capítulo

Con el desarrollo de este capítulo se han analizados los conceptos básicos relacionados con la UV, sus componentes: LMS, ROA, HA, SGA, SV, BV y LV. Algunos ejemplos de UV a lo largo del mundo tales

como: la UOC, UNED de España y UNED de Costa Rica. Se expusieron además, algunos estándares de interoperabilidad para facilitar la comunicación entre dichos componentes utilizando servicios Web a través de una Arquitectura SOA, resaltando entre ellos: IMS DRI, IMS ES, IMS QTI, IMS LIP, OKI, SQI, OAI-PMH, Z39.50 y otros. Para el envío seguro de datos fueron analizados los estándares de seguridad WS Security, XML Encryption, XML Digital Signature, SAML, XKMS, XrML y XACML.

CAPÍTULO 2. DESCRIPCIÓN DE LA SOLUCIÓN PROPUESTA

2. Introducción del Capítulo 2

En el presente capítulo se pretende brindar la propuesta y descripción de la solución definitiva para el modelo de interoperabilidad entre los componentes de la Universidad Virtual para la UCI (UVUCI). Con tales propósitos y para una mejor comprensión, se enuncian los componentes definitivos que reúne la UVUCI. Se describirá la arquitectura de la misma, así como los principales métodos que dotan de funcionalidad y operatividad a la solución propuesta a través de los estándares de interoperabilidad aplicados, se explicará el proceso a seguir para garantizar la seguridad en el envío de datos mediante la aplicación de estándares de seguridad y se mencionarán las pautas y las propuestas futuras a incorporar en la UVUCI.

2.1. Modelo de la Universidad Virtual de la UCI (UVUCI)

El modelo de UVUCI, está centrado en la selección de un conjunto de componentes que garantizan las funcionalidades básicas en una UV, que puedan interactuar de manera transparente para el usuario y sean flexibles a nivel de sistema. El estudio de otras universidades de renombre como: la UOC, la UNED de España, etc., arrojaron gran parte de los componentes, tecnologías y estándares a utilizar en la propuesta para la UVUCI. En homología con las UVs previamente mencionadas, se identificaron y trataron de aprovechar aquellos componentes en funcionamiento o en desarrollo dentro de la UCI.

Para el modelo definitivo de UVUCI fueron seleccionados los siguientes componentes: LMS, ROA, HA, BV, LV y SGA.

Es válido aclarar que a pesar de las muchas ventajas que aportaría tener en la propuesta de solución un componente como la SV, el mismo no será incluido en la propuesta final dado que sólo interesan procesos particulares de una SV tales como: Acreditación, Certificación, Titulación y no todo lo que una SV puede ofrecer o es responsable de llevar a cabo como el pago de los cursos en línea, gestión de horarios docentes, etc. En su lugar estas funciones serán gestionadas desde el nuevo SGA que se desarrollará

CAPÍTULO 2. DESCRIPCIÓN DE LA SOLUCIÓN PROPUESTA

que tendrá por nombre RP Universitario. Con esto se pretende simplificar el modelo de interoperabilidad, con una arquitectura menos dependiente y sencilla respecto al número de componentes y menos compleja.

Por otro lado, el LV a utilizar es el laboratorio remoto pues a pesar de la alta conectividad a internet que la misma demanda ofrece mayores ventajas a los usuarios. Asimismo, se utilizará en la propuesta definitiva una biblioteca electrónica por ser una extensión de las bibliotecas digitales y además añade otros servicios y funcionalidades. Convenientemente, Biblioteca Virtual o BV serán las denominaciones recibidas para las bibliotecas electrónicas en la solución final.

Para un mejor entendimiento de la propuesta para la UVUCI (Ver Figura 6), se dividirá el modelo en dos secciones: **Acceso al entorno e-learning** y **Flujo básico de eventos dentro del entorno**.

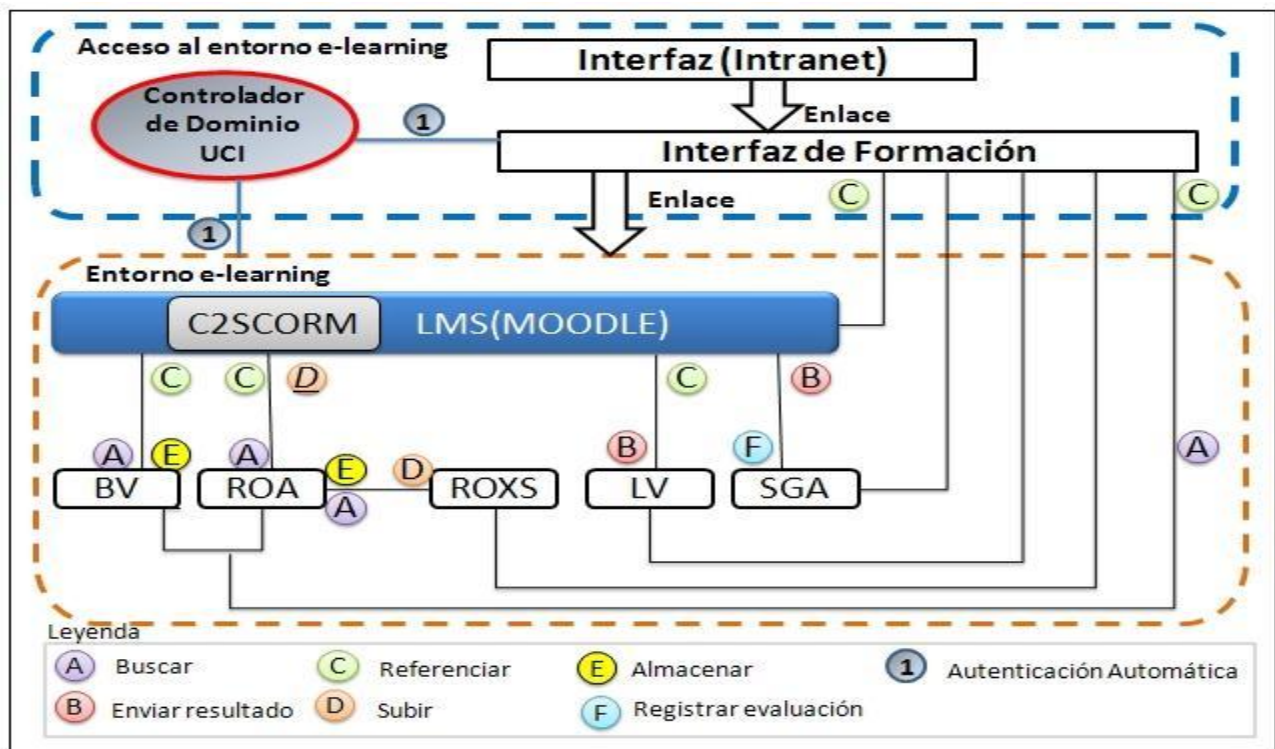


Figura 6. Modelo de la UV para la UCI

Acceso al entorno e-learning

El acceso al entorno e-learning tiene como primer paso el acceso a una interfaz general (Intranet⁸⁸) desde la cual se puede acceder a una interfaz de formación y los procesos educativos en la UCI. Dicha interfaz es el puente o vehículo fundamental para acceder a la UVUCI. Aquí se muestra información sobre el campus. Desde esta interfaz se pueden realizar búsquedas en la BV así como en el Repositorio de Objetos de Aprendizaje, los resultados obtenidos en estas búsquedas son referenciados en la interfaz, de igual modo son referenciados: cursos, maestrías, posgrados, etc. que se estén ofertando, tomados del SGA o LMS.

Referente al entorno de formación o campus virtual en cuestión, debe resaltarse que existen dos vías para acceder al mismo:

- La primera de estas vías brinda acceso como invitado, dado que el usuario puede navegar libremente por el campus, realizar búsquedas, visualizar contenidos y consumir otros servicios, pero no dispone de un acceso pleno a todas las funcionalidades del CV. A pesar de poder accederse de manera independiente al campus, el acceso estándar debe realizarse a través de la interfaz general la cual facilita acceso a una interfaz de formación que a su vez nos abre las puertas del entorno de formación.
- La otra vía de acceso es a través de un controlador de dominio que permite identificar al usuario y el rol al que pertenece, de esta manera se pueden hacer personalizaciones de las vistas, interfaces que interactúan y los servicios que puede consumir ese usuario. Por este medio se puede navegar por todo el campus con autenticación única, es decir, sin necesidad de autenticarse nuevamente en otros sistemas dentro del mismo.

⁸⁸ Es una red de ordenadores privados que utiliza tecnología Internet para compartir de forma segura cualquier información o programa del sistema operativo

Flujo básico de eventos dentro del entorno

El flujo básico de eventos dentro del entorno describe las principales relaciones y cómo interactúan los componentes que conforman dicho entorno. El LMS es el componente medular del campus dado que tiene por actividad fundamental la gestión del aprendizaje del estudiante, el resto de los componentes sirven como complemento (no por ello menos útiles e importantes) a esta actividad. A continuación serán detalladas cada una de las interacciones entre los componentes del entorno:

- La primera relación a destacar es la que se establece entre los componentes ROXS (Herramienta de Autor) y ROA (Repositorio de Objeto de Aprendizaje). ROXS es el componente encargado de la creación y edición de los OAs, para la edición de un OA almacenado en ROA, el mismo es buscado y subido por ROA a ROXS, luego de editarse es almacenado nuevamente en el repositorio. Un proceso similar ocurre en la creación de estos OAs, estos son creados en ROXS y luego almacenados en ROA.
- La segunda interacción ocurre entre el LMS y el ROA. El LMS posee una referencia de los OA existentes en ROA, asimismo puede realizar búsquedas en ROA de OAs y utilizar los contenidos almacenados. C2SCORM es uno de los módulos integrados a moodle y el mismo permite exportar los cursos contenidos en el LMS como paquetes SCORM para luego ser almacenados en el ROA, debe resaltarse que como en el LMS solo se poseen referencias a los OAs y recursos utilizados en los cursos, los mismos deben ser subidos a moodle (excepcionalmente) dado que C2SCORM para generar los paquetes SCORM necesita tener una copia temporal de los contenidos que utiliza el curso. C2SCORM, a pesar de estar capacitado para gestionar referencias a los OAs de un curso, en el momento de generar el paquete SCORM no utiliza referencias, sino los OAs en sí. Una vez generado el paquete SCORM el mismo es almacenado en ROA.
- Otra de las relaciones ocurre entre el LMS y la BV. El LMS busca información en la Biblioteca y son referenciada en el LMS los resultados de estas búsquedas así como recursos de la biblioteca.
- Además se incluye la comunicación con el SGA para que moodle pueda registrar las evaluaciones realizadas por los estudiantes y a su vez tomar la estructura de los grupos docentes.
- La última interrelación representada es entre el LV y el LMS, en el cual el LMS referencia la práctica de laboratorio existentes y el LV a su vez envía los resultados o los cuestionarios realizados en las prácticas, hacia el LMS.

CAPÍTULO 2. DESCRIPCIÓN DE LA SOLUCIÓN PROPUESTA

No debe omitirse, que cada una de las herramientas y sistemas que se integran dentro del modelo de UVUCI a pesar de trabajar relacionadas, tienen puntos independientes de acceso y están capacitadas para brindar sus servicios sin afectar al resto de los componentes, así como colaborar con los mismos. Se puede apreciar entre otras cosas el grado de flexibilidad que poseen la arquitectura concebida en la solución. En el siguiente apéndice se describe el modelo arquitectónico general de la UVUCI a través de una arquitectura en capas.

2.2. Arquitectura de la UVUCI

La arquitectura de la UVUCI descansa y tiene por basamento el FI propuesto en [4]. En el “Framework Integrador para la interoperabilidad entre herramientas que gestionan contenidos educativos reutilizables”, se logra la interoperabilidad entre los componentes ROA, ROXS y el LMS moodle a partir de estándares que facilitan la comunicación y el intercambio entre diferentes herramientas. Este FI propone en su solución una arquitectura orientada a servicios (SOA).

En la siguiente Figura, se muestra una arquitectura en capas que refiere al esquema general e ideal para aplicaciones desarrolladas desde cero con SOA. Se presenta una arquitectura base, compuesta por tres capas: la de Presentación, la de Servicios (Generales y de Interoperabilidad) y la de Acceso a Datos.



Figura 7. *Arquitectura de la UV para la UCI*

CAPÍTULO 2. DESCRIPCIÓN DE LA SOLUCIÓN PROPUESTA

La capa de **Presentación** es la capa a través de la cual interactúa el usuario de la aplicación con la misma. Esta capa maneja todo lo referente al usuario y es la mediadora a la capa de servicios.

La capa de **Servicios** incluye tanto servicios generales⁸⁹ como los servicios que posibilitan la interoperabilidad entre los diferentes componentes de la UV, manejando todos los procesos o la lógica del negocio del sistema. Según [4] la responsabilidad de la misma es implementar lógica de negocio que será consumida por la interfaz del sistema y por otros sistemas a través de los servicios compartidos. Dichos servicios se basan en un conjunto de estándares de comunicación, como son XML para la representación de datos, SOAP para el intercambio de datos y el lenguaje WSDL para describir las funcionalidades de un servicio web.

La tercera y última capa de la arquitectura, la de **Acceso a Datos** es accedida desde el BUS de servicios residente en la capa inmediata superior, proporcionando el enlace de la capa de servicios con los datos que ésta necesita para funcionar.

En la subcapa de **Servicios de Interoperabilidad** como se puede ver en la Figura 8, están contemplados los componentes ROXS, ROA, SGA, BV, LV y el LMS moodle (definidos para el modelo de la UV en la sección anterior), que son herramientas que por su complejidad, tamaño y el coste que representaría desarrollarlas desde cero siguiendo una arquitectura orientada a servicios se opta por no cambiar la arquitectura de los mismos y en su lugar se proveen mediante pasarelas o gateways de comunicación, los mecanismos necesarios para llevar a cabo la traducción de las peticiones y las repuestas de estos componentes hacia y desde las capas inferiores en donde residen el Bus de Servicios y otros componentes. Estas pasarelas incluyen un conjunto de APIs⁹⁰ que proporcionen la interpretación de los servicios.

⁸⁹ Servicios adicionales y específicos de cada herramienta proveídos de manera independiente.

⁹⁰ Application Programming Interface.

CAPÍTULO 2. DESCRIPCIÓN DE LA SOLUCIÓN PROPUESTA

El LMS se encuentra en el extremo inferior separado de los demás componentes precisamente porque el modelo se basa en la comunicación de los otros componentes con el LMS Moodle que es quien guía el proceso de aprendizaje del estudiante.

Todos estos componentes contienen un conjunto de estándares (SCORM, IMS CP, Marc 21) que no son para la integración de herramientas pero favorecen el intercambio de información o de contenido entre las diferentes aplicaciones, por ejemplo el componente ROXS como se vio en el Capítulo 1 incluye el estándar SCORM con el fin de empaquetar los contenidos para importarse y exportarse entre plataformas, compartir, reutilizar, importar y exportar OAs y el IMS CP para describir y empaquetar contenidos de forma que puedan ser procesados por otro LMS diferente. Para el caso del ROA contempla el SCORM e IMS CP, en el SGA no contempla ningún estándar, la BV contempla el Marc 21 para la catalogación de los recursos, el LV utiliza el IMS QTI para el intercambio de cuestionarios con la plataforma⁹¹ y el LMS incluye el SCORM, IMS ES y el IMS QTI.



Figura 8. Subcapa Servicio de Interoperabilidad de la Arquitectura de la UVUCI

⁹¹ Se refiere al LMS.

CAPÍTULO 2. DESCRIPCIÓN DE LA SOLUCIÓN PROPUESTA

En el Bus de Servicios (Ver figura 9.) se inserta una capa de Servicios de OKI (OSID) que permite usar en forma de servicios web algunas de las OSIDs básicas (Autenticación, Autorización y Logging) de OKI para conectar las aplicaciones. Se adhieren a este bus de servicios los estándares SQI para Solicitar/Entregar y Buscar/Entregar en ROXS y ROA, el IMS DRI para Enviar/Almacenar en ROA, el Z39.50 para búsqueda y recuperación de información en la BV, el IMS LIP para definir un conjunto de paquetes que pueden ser usados para importar, extraer e intercambiar datos de estudiantes desde el punto de vista educativo entre el SGA y LMS, el IMS ES puede utilizarse para especificar matrículas / anulaciones en los cursos, información del curso, información de usuario, mantenimiento de expedientes y administración de grupos desde el punto de vista administrativo en el SGA y por último la capa de los Estándares de Seguridad, incluyendo WS-Security, XML Encryption, XML Digital Signature y SAML para que se realice la transferencia de información de forma segura.



Figura 9. Bus de Servicios

Como es apreciable a grandes rasgos, el lograr la interoperabilidad entre estas herramientas no sería tarea fácil sin la inclusión de estándares en la solución propuesta. Los estándares son los pilares principales que permiten la comunicación entre las aplicaciones que conforman la UVUCI.

CAPÍTULO 2. DESCRIPCIÓN DE LA SOLUCIÓN PROPUESTA

A continuación se hace una explicación detallada de los mismos, algunos de ellos ya fueron vistos en el capítulo 1.

2.3. Descripción de Estándares utilizados en la UVUCI

En la arquitectura de la UVUCI (como se vio anteriormente) se utilizan de los estándares propuestos por el FI a: SQI, IMS-DRI así como las guías OSID de OKI: Authentication y Authorization. Se determinó no utilizar la osid Repository pues la interoperabilidad con el repositorio ya se tiene en cuenta con IMS DRI y SQI y para el CV inicialmente existe un solo ROA. Son añadidos al modelo UVUCI además de los estándares antes mencionados: Z39.50, IMS ES, IMS LIP y la osid Logging. A continuación será descrita la forma de utilizar estos estándares.

2.3.1. Descripción de los métodos de Simple Query Interface (SQI)

Este estándar como se ha dicho antes fue propuesto por [4] en el “Framework Integrador para la interoperabilidad entre herramientas que gestionan contenidos educativos reutilizables” para búsquedas de contenidos digitales en ROA.

A continuación se describirán los 10 métodos de SQI teniendo en cuenta una breve descripción basándose en la especificación de SQI de [36]:

Método	Descripción
SetQueryLanguage	Este método permite a la fuente tener el control de la sintaxis usada para identificar el lenguaje de la consulta. Los valores para el parámetro queryLanguageID se distingue entre mayúsculas y minúsculas.
SetMaxQueryResults	Este método define el número máximo de resultados, que puede producir una consulta. El máximo se fija en 100 por defecto, pero se puede controlar por medio del parámetro maxQueryResults(), que debe ser cero (0) o mayor. Si se pone en cero (0) el número máximo de consultas, la fuente acepta todos los resultados producidos por la misma.

CAPÍTULO 2. DESCRIPCIÓN DE LA SOLUCIÓN PROPUESTA

setMaxDuration	Este método permite a la fuente establecer un tiempo de consulta en caso de una interfaz de consulta asincrónica. Los valores de maxDuration deben ser cero (0) o mayor, y son interpretados en milisegundos, si no se define un valor, toma por defecto cero (0).
setResultsFormat	Este método permite a la fuente tener el control del formato de resultado retornado por el destino y que es especificado por el parámetro resultsFormat.
setResultsSetSize	Este método define el número máximo de resultados. El tamaño de los resultados está establecido por 25 registros por defecto, pero puede ser controlado por el método resultsSetSize que puede ser cero (0) o mayor.
synchronousQuery	Este método envía una consulta al destino conformado por varios parámetros, uno de ellos es queryStatement, el cual contiene el cuerpo de la consulta, y targetSessionID que identifica la sesión, el método devuelve los resultados dependiendo de la cantidad de coincidencias encontradas. El parámetro startResult identifica el comienzo de una serie de resultados, y que éste debe estar entre el rango 1 y el número total de resultados que se define por el método setResultsSetSize ⁹² , y el máximo permitido está dado por setMaxQueryResults.
getTotalResultsCount	Este método retorna el número total de resultados disponibles por una consulta. Se le pasa como parámetro el targetSessionID para identificar la sesión y el cuerpo de la consulta a través del parámetro queryStatement.
setSourceLocation	Este método es requerido antes de que se produzca una consulta asíncrona. El parámetro sourceLocation especifica la localización del listener ⁹³ de la fuente para que el destino pueda ubicarlo, esto es a través de una URL.
asynchronousQuery	Este método permite a la fuente enviar una consulta al destino, mientras los resultados son retornados en un modo asíncronico. El cuerpo de la consulta

⁹² Visto anteriormente.

⁹³ Un escuchador que espera por los resultados.

CAPÍTULO 2. DESCRIPCIÓN DE LA SOLUCIÓN PROPUESTA

	<p>se especifica por el parámetro <code>queryStatement</code>. Además se le pasa un identificador de consulta que es emitido por la fuente y se requiere para hacer un vínculo de los resultados de la consulta, a la consulta cuando éstos son más tarde retornados usando el <code>results listener</code>. Mediante el uso de un identificador de las consultas es posible enviar un número arbitrario de consultas por cada sesión activa. La localización del listener en la fuente se necesita y se debe proveer usando el método <code>setSourceLocation</code>. Debido a la naturaleza asincrónica de este método, pueden llegar resultados de consultas anteriores. La consulta se procesa y los resultados se envían en el plazo señalado en el método <code>setMaxDuration</code>.</p>
queryResultsListener	<p>Este método, iniciado por el destino, devuelve los resultados hacia la fuente. El parámetro <code>queryID</code> es usado para referenciar los resultados con la previa consulta enviada, cuando éstos son retornados más tarde usando el <code>results listener</code>. El <code>queryResults</code> tiene un conjunto de resultados que consiste en una lista de registros de metadatos, los cuales concuerdan con el esquema especificado en la consulta.</p>
CreateSession	<p>Este método crea la sesión, para ello requiere los parámetros <code>userID</code> and <code>password</code>, retornando un <code>Session ID</code>.</p>
CreateAnonymousSession	<p>Este método crea una sesión sin necesidad de una cuenta en el sistema, el mismo devuelve un ID de la sesión.</p>
DestroySession	<p>Este método destruye la sesión, se identifica la misma a través del parámetro <code>sessionID</code>.</p>

Tabla 4. Métodos de SQL.

2.3.2.Las guías OSID de OKI.

Para la UVUCI se tiene en cuenta diferentes OSID de OKI, como la Authentication, Authorization y Logging las cuales contienen un conjunto de interfaces y métodos que se explicará en los próximos acápite.

CAPÍTULO 2. DESCRIPCIÓN DE LA SOLUCIÓN PROPUESTA

2.3.2.1. Métodos de la OSID Authentication

En capítulo anterior se vio que la OSID Authentication les proporciona a los consumidores adquirir y validar las credenciales de autenticación sin tener que gestionar los detalles de un entorno de autenticación. Soporta invocar procesos de autenticación y la implementación de estos servicios son responsables de garantizar cualquier información apropiada para llevar a cabo este proceso.

A continuación se explicarán algunos métodos que se deben implementar para la UVUCI. Estos métodos⁹⁴ fueron definidos por [51].

Método	Descripción
GetAuthenticationTypes()	Este método devuelve los tipos de autenticación que son soportados por la implementación.
GetAuthenticationTypes	Este método devuelve los tipos de autenticación que son soportados por la implementación.
authenticateUser	Invoca el proceso de autenticación del tipo especificado para identificar al usuario. Puede ser necesario llamar a <code>isUserAuthenticated ()</code> para comprobar el estado de autenticación. Este estándar limita el tiempo de autenticación de los usuarios; para ello requiere consultas explícitas del estado de autenticación. El control del estado de autenticación se producirá con más frecuencia que invocar el mecanismo para autenticar al usuario.
destroyAuthentication	Destruye la autenticación para todos los tipos de autenticación.
destroyAuthenticationForType	Destruye la autenticación para los tipos de autenticaciones especificadas.
getUserId	Retorna un único ID del agente que representa al usuario para un tipo de autenticación específica. Estos agentes son administrados en la Shared OSID ⁹⁵ .

⁹⁴ En algunos de los casos se mantendrán palabras en inglés para no alterar su significado.

⁹⁵ Otra OSID que puede ser consultada en <http://www.okiproject.org/>.

CAPÍTULO 2. DESCRIPCIÓN DE LA SOLUCIÓN PROPUESTA

isUserAuthenticated	Comprueba el estado actual de autenticación del usuario. Si retorna verdadero, el usuario fue autenticado y si retorna falso no logró dicha autenticación. Esto último puede indicar que los usuarios no pueden ser autenticados o que se ha caducado su cuenta.
----------------------------	--

Tabla 5. Métodos de la osid autenticación

2.3.2.2. Métodos de la OSID Authorization

La OSID Authorization, como se evidenció en el capítulo 1, provee una forma de definir quién está autorizado para hacer qué y cuándo, asociando las autorizaciones de agentes⁹⁶, que representan el usuario u otro actor en el sistema, con funciones y Calificadores⁹⁷.

A continuación se explicarán algunos métodos que fueron definidos por [52] y que son propuestos para que se implementen en la UVUCI.

Método	Descripción
agentExists	Retorna verdadero si el agentId existe en el servicio de autorización y falso en caso contrario.
createAuthorization	Crea una nueva autorización para que un agente realice funciones con un calificador ⁹⁸ . Usa la fecha y el tiempo actual como la effectiveDate y no modifica la fecha de expiración.
createDatedAuthorization	Crea una nueva autorización para que un agente realice una función con un calificador.
createFunction	Los IDs en Autorización son definidos en el exterior y su singularidad es forzada por la aplicación.
createQualifier	Los IDs en autorización son definidos en el exterior y su singularidad es forzada por la aplicación. Crea un nuevo calificador en el servicio

⁹⁶ Una representación de un director o actor conocido a un sistema.

⁹⁷ Es un identificador para el objeto de la acción de una función. Un Calificador tiene un y sólo un Tipo de Calificador.

⁹⁸ Un identificador para el objeto de una acción de una Function. Un calificador tiene uno y sólo un QualifierType.

CAPÍTULO 2. DESCRIPCIÓN DE LA SOLUCIÓN PROPUESTA

	de autorización. Esto es diferente a hacer una nueva instancia de un calificador local ya que el calificador será insertado en el servicio de autorización.
createRootQualifier	Crea un nuevo calificador en el servicio de autorización que no tiene padres. Esto es diferente de la realización de una nueva instancia de un objeto clasificador a nivel local ya que este será insertado en los servicios de autorización.
deleteAuthorization	Elimina una autorización existente.
deleteFunction	Elimina una function ⁹⁹ por ID.
deleteQualifier	Elimina un calificador por Id. El calificador raíz no puede ser eliminado.
getAllAZs	Dada una functionId y un calificador devuelve todas las autorizaciones que permitirán a los agentes hacer la function con el calificador. Este método difiere de getAuthorizations en que busca cualquier autorización que permite hacer la function con el calificador, incluso si el calificador de autorización pasa a ser un padre de este argumento clasificador.
getAllAZsByFuncType	Dada una FunctionType y un calificador devuelve todas las autorizaciones que permitan a los agentes hacer función en el FunctionType con el calificador. Este método difiere de getAuthorizations en que este método busca cualquier autorización que permite hacer la function con el calificador, incluso si el calificador de la autorización pasa a ser un padre del argumento clasificador.
getAllUserAZs	Dada una functionId y un calificador devuelve todas las autorizaciones que permiten al usuario hacer la función con el calificador. Este método difiere de la simple forma de

⁹⁹ Una acción o rol que uno puede ser autorizado a tomar. Una función tiene una y solo una FunctionType.

CAPÍTULO 2. DESCRIPCIÓN DE LA SOLUCIÓN PROPUESTA

	<p>getAuthorizations en que este método busca cualquier autorización que permite hacer la función con el calificador, incluso si el calificador de la autorización pasa a ser un padre de este argumento calificador.</p>
getAllUserAZsByFuncType	<p>Dada una FunctionType y un calificador devuelve todas las autorizaciones que permiten al usuario realizar las functions en el FunctionType con el calificador. Este método difiere de getAuthorizations en que este método busca cualquier autorización que permite hacer la función con el calificador, incluso si el calificador de la autorización pasa a ser un padre del argumento calificador.</p>
getExplicitAZs	<p>Dado un agentId, un functionId, y un qualifierId (al menos uno de estos debe ser no nulo) devuelve las correspondientes autorizaciones.</p>
getExplicitAZsByFuncType	<p>Dado un agentId, un FunctionType, y un calificador (ya sea agentId o qualifierId debe ser no nulo) devuelve las correspondientes autorizaciones. Las autorizaciones deben ser para las funciones dentro de los FunctionType dado.</p>
getExplicitUserAZs	<p>Dado un functionId y calificador (uno no debe ser no nulo) devuelve las correspondientes autorizaciones de usuarios.</p>
getExplicitUserAZsByFuncType	<p>Dado un FunctionType y calificador devuelve las correspondientes autorizaciones de usuarios. Las autorizaciones deben ser para las funciones dentro de los FunctionType dado.</p>
getExplicitUserAZsForImplicitAZ	<p>Dado un implícito¹⁰⁰ devuelve las correspondientes autorizaciones explícitas del usuario. Autorizaciones explícitas pueden ser modificadas.</p>

¹⁰⁰ Una autorización que no fue creada utilizando AuthorizationManager y que tiene el agente, función, y calificador o calificador Descendiente (véase jerarquía calificador) de una autorización explícita. La autorización implícita, a diferencia de la explícita, no se puede modificar directamente. Pueden ser modificadas sólo a través de una autorización explícita que contiene un ancestro calificador.

CAPÍTULO 2. DESCRIPCIÓN DE LA SOLUCIÓN PROPUESTA

getFunction	Devuelve todas las funciones.
getFunctions	Devuelve todas las funciones dado un tipo específico.
getFunctionTypes	Obtener todos los FunctionTypes soportados por la implementación de autorización.
getQualifier	Retorna los calificadores existentes
getQualifierChildren	Dado un qualifierId existente retorna los Ids de sus calificadores hijos.
getQualifierDescendants	Dado un qualifierId existente retorna los Ids de todos los descendientes incluyendo los calificadores de su hijo.
getQualifierHierarchies	Devuelve las Qualifier Hierarch ¹⁰¹ soportadas por la implementación de autorización. Las Qualifier Hierarchy son referenciadas por Id y pueden ser conocida y administradas a través de la Jerarquía OSID ¹⁰² .
getQualifierTypes	Retorna todos los QualifierTypes soportados por la implementación de autorización.
getRootQualifiers	Dado un hierarchyId, devuelve los calificadores de la raíz de la jerarquía especificada.
getWhoCanDo	Dada una functionId y un calificador devuelve el ID de todos los agentes autorizados a hacer la función con el calificador. Un calificador nulo es tratado como un comodín ¹⁰³ .
isAuthorized	Dado un agentId, functionId, y calificador devuelve verdadero si el agente está autorizado actualmente para realizar la función con el calificador.
isUserAuthorized	Dada una functionId y calificador devuelve verdadero si el usuario está autorizado actualmente para realizar la función con el

¹⁰¹ Una organización de los calificadores en los nodos de tal forma que todas las autorizaciones concedidas para un ancestro nodo se conceden a todos los nodos descendientes.

¹⁰² Consultar www.okiproject.org

¹⁰³ Puede ser cualquier calificador.

CAPÍTULO 2. DESCRIPCIÓN DE LA SOLUCIÓN PROPUESTA

	calificador.
supportsDesign	Este método indica si esta implementación soporta los métodos AuthorizationManager: createFunction (), deleteFunction (). Métodos Función: UpdateDescription (), updateDisplayName ().
supportsMaintenance	Este método indica si esta implementación soporta los métodos AuthorizationManager:createAuthorization,createDatedAuthorization, createQualifier, createRootQualifier, deleteAuthorization, deleteQualifier, getFunctionTypes, getQualifier, getQualifierChildren, getQualifierDescendents, GetQualifierHierarchies, getQualifierTypes, getRootQualifiers, getWhoCanDo. Función métodos: getDescription, getDisplayName, getFunctionType, getId, GetQualifierHierarchy. Calificador métodos: addParent, changeParent, getChildren, getDescription, getDisplayName, getId, isParent, getParents, getQualifierType, isChildOf, isDescendentOf, removeParent, updateDescription, updateDisplayName.

Tabla 6. Métodos de la osid Authorization

2.3.2.3. Métodos de la OSID Logging

La OSID Logging será utilizada con el objetivo de proporcionar un medio que permita la lectura y escritura de registros. Un registro puede representar a una colección de registros de entrada. Así mismo cada uno de estos registros se encuentra compuesto por un tipo de prioridad, fecha y hora, agente y el contenido escrito. Igualmente estos registros pueden ser organizados en federaciones de jerarquías pudiendo hacer visibles las entradas de los registros hijos a través de los registros de sus correspondientes padres.

A continuación serán descritos los métodos que se utilizarán de esta OSID según [53], los cuales se encuentran agrupados en las interfaces: *osid.logging.LoggingProfile*, *osid.logging.LoggingSession*, *osid.logging.LogEntry* y *osid.logging.Log*

CAPÍTULO 2. DESCRIPCIÓN DE LA SOLUCIÓN PROPUESTA

- **Interface osid.logging.LoggingProfile:** Esta interfaz describe el perfil de la interoperabilidad entre los servicios de registro.

Método	supportsVisibleFederation()
Descripción	Prueba o verifica si la visibilidad de federación es soportada.
Tipo de Retorno	Boolean (true si la federación es visible, false en otro caso)

Tabla 7. Método supportsVisibleFederation de la OSID Logging

Método	supportsLogging()
Descripción	Prueba si la entrada de registros es soportada
Tipo de Retorno	Boolean (true si los registros de lectura son soportados, false en otro caso)

Tabla 8. Método supportsLogging de la OSID Logging

Método	supportsLogReading()
Descripción	Prueba si la lectura de registros es soportada.
Tipo de Retorno	Boolean (true si la lectura de registros es soportada, false en otro caso)

Tabla 9. Método supportsLogReading de la OSID Logging

Método	supportsLogEntrySearch()
Descripción	Prueba si la búsqueda de entradas de registros es soportada
Tipo de Retorno	Boolean (true si la búsqueda de entradas de registros es soportada, false en otro caso)

Tabla 10. Método supportsLogEntrySearch de la OSID Logging

Método	supportsLogEntryNotification()
Descripción	Prueba si la notificación de entrada de registros es soportada
Tipo de Retorno	Boolean (true si la notificación de entrada de registros es soportada, false en otro caso)

Tabla 11. Método supportsLogEntryNotification de la OSID Logging

CAPÍTULO 2. DESCRIPCIÓN DE LA SOLUCIÓN PROPUESTA

Método	supportsLogLookup()
Descripción	Prueba la disponibilidad de un servicio de registro de búsqueda.
Tipo de Retorno	Boolean (true si existe un registro de búsqueda, false en otro caso)

Tabla 12. Método supportsLogLookup de la OSID Logging

Método	supportsLogSearch()
Descripción	Prueba si la búsqueda de registros está disponible
Tipo de Retorno	Boolean (true si la búsqueda de registros está disponible, false en otro caso)

Tabla 13. Método supportsLogSearch de la OSID Logging

Método	supportsLogAdmin()
Descripción	Prueba la disponibilidad de un registro administrativo de servicios para crear y eliminar registros.
Tipo de Retorno	Boolean (true si un registro administrativo está disponible, false en otro caso)

Tabla 14. Método supportsLogAdmin de la OSID Logging

Método	supportsLogNotification()
Descripción	Prueba la disponibilidad de un servicio de notificación de registro.
Tipo de Retorno	Boolean (true si hay disponible un servicio de notificación de registro, false en otro caso)

Tabla 15. Método supportsLogNotification de la OSID Logging

Método	supportsLogHierarchy()
Descripción	Prueba la disponibilidad de un servicio de registro de jerarquía transversal.
Tipo de Retorno	Boolean (true si existe este tipo de servicio, false en otro caso)

Tabla 16. Método supportsLogHierarchy de la OSID Logging

Método	supportsLogHierarchyDesign()
---------------	-------------------------------------

CAPÍTULO 2. DESCRIPCIÓN DE LA SOLUCIÓN PROPUESTA

Descripción	Prueba la disponibilidad de un servicio de diseño de jerarquía de registro
Tipo de Retorno	Boolean (true si existe este servicio, false en otro caso)

Tabla 17. Método supportsLogHierarchyDesign de la OSID Logging

Método	supportsLogHierarchySequencing()
Descripción	Prueba si la jerarquía de registro soporta nodos de secuencia
Tipo de Retorno	Boolean (true si es soportado, false si no)

Tabla 18. Método supportsLogHierarchySequencing de la OSID Logging

Método	getLogEntrySearchRecordTypes()
Descripción	Obtiene los tipos de registro en la entrada de un registro de búsqueda
Tipo de Retorno	osid.type.TypeList (una lista que contiene los tipos de registro en la entrada de un registro de búsqueda)

Tabla 19. Método getLogEntrySearchRecordTypes de la OSID Logging

Método	supportsLogEntrySearchRecordType()
Descripción	Prueba si el tipo de registro en la entrada de un registro de búsqueda es soportado
Tipo de Retorno	Boolean (true si el tipo de registro dado es soportado, false si no)
Parámetros	osid.type.Type logEntrySearchRecordType (Un tipo indicando el tipo de registro)

Tabla 20. Método supportsLogEntrySearchRecordType de la OSID Logging

Método	getLogRecordTypes()
Descripción	Obtiene los tipos de registro soportados
Tipo de Retorno	osid.type.TypeList (una lista que contiene los tipos de datos soportados)

Tabla 21. Método getLogRecordTypes de la OSID Logging

Método	supportsLogRecordType()
Descripción	Prueba si un tipo de registro dado es soportado.

CAPÍTULO 2. DESCRIPCIÓN DE LA SOLUCIÓN PROPUESTA

Tipo de Retorno	Boolean (true si el tipo de registro dado es soportado, false si no)
Parámetros	osid.type.Type logRecordType (Un tipo indicando el tipo de registro)

Tabla 22. Método supportsLogRecordType de la OSID Logging

Método	getLogSearchRecordTypes()
Descripción	Obtiene los tipos de datos soportados por los registros de búsqueda.
Tipo de Retorno	osid.type.TypeList (una lista que contiene el registro de búsqueda que soporta los tipos de registro)

Tabla 23. Método getLogSearchRecordTypes de la OSID Logging

Método	supportLogSearchRecordType()
Descripción	Prueba si el tipo de registro es soportado en la búsqueda de registro.
Tipo de Retorno	Boolean (true si el tipo dado es soportado, false en otro caso)
Parámetros	osid.type.Type logSearchRecordType (Un tipo de registro que indica un tipo de registro)

Tabla 24. Método supportLogSearchRecordType de la OSID Logging

Método	getPriorityTypes()
Descripción	Obtiene los tipos de prioridad soportados, en orden ascendente al nivel de prioridad
Tipo de Retorno	osid.type.TypeList (retorna una lista que contiene los tipos de prioridad soportados)

Tabla 25. Método getPriorityTypes de la OSID Logging

Método	supportsPriorityType()
Descripción	Prueba si la prioridad de tipos es soportada
Tipo de Retorno	Boolean (true si un tipo dado es soportado, false en otro caso)
Parámetros	osid.type.Type priorityType (Un tipo indicando el tipo de prioridad)

Tabla 26. Método supportsPriorityType de la OSID Logging

CAPÍTULO 2. DESCRIPCIÓN DE LA SOLUCIÓN PROPUESTA

Método	getContentTypes()
Descripción	Obtiene el registro de entrada de los tipos de contenido soportados
Tipo de Retorno	osid.type.TypeList (retorna una lista que contiene el registro de entrada soportado para un tipo de contenido)

Tabla 27. Método getContentTypes de la OSID Logging

Método	supportsContentType()
Descripción	Prueba si la entrada de registros de tipos de contenidos es soportada
Tipo de Retorno	Boolean (true si un tipo dado es soportado, false en otro caso)
Parámetros	osid.type.Type contentType (un tipo indicando un tipo de contenido para la entrada de registro)

Tabla 28. Método supportsContentType de la OSID Logging

- **Interface osid.logging.LoggingSession:** Esta sesión se utiliza para hacer entradas de registro a un registro.

Método	getLogId()
Descripción	Obtiene el Id de registro asociado a la sesión.
Tipo de Retorno	osid.id.Id ()

Tabla 29. Método getLogId de la OSID Logging

Método	getLog()
Descripción	Obtiene el registro asociado
Tipo de Retorno	osid.logging.Log (El registro)
Errores	OPERATION_FAILED (incapaz de completar la solicitud) PERMISSION_DENIED (autorización fallida)

Tabla 30. Método getLog de la OSID Logging

Método	canLog()
---------------	-----------------

CAPÍTULO 2. DESCRIPCIÓN DE LA SOLUCIÓN PROPUESTA

Descripción	Pruebas si este usuario puede iniciar sesión. El retorno de éxito (true) no garantiza la autorización. El retorno de falso (false) indica que todos los métodos en esta sesión tendrá como resultado una PERMISSION_DENIED (denegación de permisos). Esto es como una pista para una aplicación que pueden optar por no ofrecer a las operaciones de registro.
Tipo de Retorno	Boolean (true si los métodos de registro no están autorizados, false en otro caso)

***Tabla 31.** Método `canLog` de la `OSID Logging`*

Método	log()
Descripción	Registra un elemento.
Parámetros	object content (La entrada al registro) osid.type.Type contentType (Este tipo de entrada debe ser uno de los tipos retornados por <code>LoggingManager.getContentType()</code>)
Errores	<code>INVALID_ARGUMENT</code> (contenido no es de <code>ContentType</code>) <code>NULL_ARGUMENT</code> (contenido o argumento siempre <code>ContentType</code>) <code>OPERATION_FAILED</code> (incapaz de completar la solicitud) <code>PERMISSION_DENIED</code> (autorización fallida) <code>UNSUPPORTED</code> (es falso)

***Tabla 32.** Método `log` de la `OSID Logging`*

Método	logAtPriority()
Descripción	Registra un elemento con un nivel de prioridad dado.
Parámetros	osid.type.Type priority (la prioridad del elemento) object content (la entrada al registro) osid.type.Type contentType (Este tipo de entrada debe ser uno de los tipos retornados por <code>LoggingManager.getContentType()</code>)
Errores	<code>INVALID_ARGUMENT</code> (contenido no es de <code>ContentType</code>) <code>NULL_ARGUMENT</code> (contenido o argumento siempre <code>ContentType</code>) <code>OPERATION_FAILED</code> (incapaz de completar la solicitud)

CAPÍTULO 2. DESCRIPCIÓN DE LA SOLUCIÓN PROPUESTA

	PERMISSION_DENIED (autorización fallida) UNSUPPORTED (es falso)
--	--

Tabla 33. Método `logAtPriority` de la OSID Logging

- **Interface `osid.logging.LogEntry`:** Esta interfaz permite trabajar con entradas de registros. Una entrada de registro se compone de un tiempo, un agente, y de algunos contenidos.

Método	<code>getId()</code>
Descripción	Se asocia con el Id asociado a esta instancia de entrada de registro. La persistencia de cualquier referencia a este objeto es realizada por la persistencia del id devuelto de este método. El Id devuelto puede ser diferente al Id utilizado para consultar este objeto. En este caso, el nuevo Id debe ser preferible sobre las consultas anteriores para las futuras consultas.
Tipo de Retorno	<code>osid.id.Id</code> (el id)

Tabla 34. Método `getId` de la OSID Logging

Método	<code>getPriority()</code>
Descripción	Obtiene el nivel de prioridad de esta entrada.
Tipo de Retorno	<code>osid.type.Type</code> (el nivel de prioridad)

Tabla 35. Método `getPriority` de la OSID Logging

Método	<code>getTimestamp()</code>
Descripción	Obtiene la fecha en que esta entrada fue registrada
Tipo de Retorno	<code>osid.calendar.Date</code> (la marca de tiempo de esta entrada)

Tabla 36. Método `getTimestamp` de la OSID Logging

Método	<code>getAgentId()</code>
Descripción	Obtiene el ID del agente que ha creado esta entrada.
Tipo de Retorno	<code>osid.id.Id</code> (el agente id)

CAPÍTULO 2. DESCRIPCIÓN DE LA SOLUCIÓN PROPUESTA

Tabla 37. Método `getAgentId` de la *OSID Logging*

Método	<code>getAgent()</code>
Descripción	Obtiene el agente que ha creado esta entrada
Tipo de Retorno	<code>osid.authentication.Agent</code> (el agente)
Errores	<code>OPERATION_FAILED</code> (Incapaz de completar la solicitud) <code>PERMISSION_DENIED</code> (Autorización fallida)

Tabla 38. Método `getAgent` de la *OSID Logging*

Método	<code>getContentType()</code>
Descripción	Obtiene el tipo aplicado por la entrada de contenido. El Tipo indica expresamente la especificación del objeto e implícitamente puede definir una familia de objetos. El tipo de interfaz puede ser devuelto como un hijo en un tipo de jerarquía. La interoperabilidad con la de interfaz tipada a los contenidos debe realizarse a través de <code>implementsContentType ()</code> .
Tipo de Retorno	<code>osid.type.Type</code> (El tipo implementado en el contenido)

Tabla 39. Método `getContentType` de la *OSID Logging*

Método	<code>hasContentType()</code>
Descripción	Prueba si el contenido soporta el tipo dado. El tipo puede ser soportado por el contenido a través de interfaz y tipo de herencia. Este método se debe comprobar antes de recuperar el tipo de extensión de la interfaz.
Tipo de Retorno	Boolean (true si el contenido implementa el tipo dado, false en otro caso)
Parámetros	<code>osid.type.Type</code> <code>contentType</code> (un tipo)
Errores	<code>NULL_ARGUMENT</code> (<code>contentType</code> is nulo)

Tabla 40. Método `hasContentType` de la *OSID Logging*

- **Interface `osid.logging.Log`:** Un registro representa una colección de entradas. Al igual que todos los objetos OSID, un registro se identifica por su Id y cualquier referencia persistente debe utilizar el id.

CAPÍTULO 2. DESCRIPCIÓN DE LA SOLUCIÓN PROPUESTA

Método	getLogRecord()
Descripción	Obtiene el registro correspondiente al tipo de registro a registro. Este método debe utilizarse para recuperar un objeto implementando la solicitud del registro de interfaces junto con todas las interfaces de su padre. El logRecordType puede ser el Tipo devuelto en getRecordTypes () o cualquiera de sus padres en un tipo de jerarquía donde hasRecordType (logRecordType) es cierto.
Tipo de Retorno	osid.logging.Log (el log de registro)
Parámetros	osid.type.Type logRecordType (el tipo de log de registro a recuperar)
Errores	NULL_ARGUMENT (logRecordType es null) OPERATION_FAILED (incapaz de completar la solicitud) PERMISSION_DENIED (autorización fallida ocurrida) UNSUPPORTED (hasRecordType(logRecordType) es falso)

Tabla 41. Método *getLogRecord* de la *OSID Logging*

2.3.3. Descripción de los métodos de IMS Enterprise Services (IMS ES)

La especificación IMS Enterprise Services como se vio en el capítulo 1, define cómo gestionar los sistemas de intercambio de información que describe personas, grupos y memberships en el contexto del aprendizaje. A continuación se explican algunos métodos a desarrollar de esta especificación definidos por [32]:

- **Interfaz PersonManager:** Es responsable de manipular la información sobre las personas. Esta interfaz apoya la manipulación de un solo objeto persona y cuenta con los siguientes métodos:

Métodos	Descripción
createPerson	Para solicitar la creación de un registro «persona» en el sistema de destino y la fuente es responsable de la asignación del identificador único
deletePerson	Para solicitar la supresión de un registro «persona». El registro «persona» es

CAPÍTULO 2. DESCRIPCIÓN DE LA SOLUCIÓN PROPUESTA

	eliminado y todas las relaciones de sus asociados
readPerson	Para leer todo el contenido identificado en el registro 'persona'. El objetivo debe devolver todos los datos que ha identificado al registro 'persona'. El objetivo debe devolver todos los datos que ha identificado para el registro 'persona'
updatePerson	Para escribir nuevos contenidos dentro del registro 'persona' identificado. El objetivo debe escribir los nuevos datos dentro del registro "persona". Esta operación es un aditivo.
replacePerson	Para reemplazar el contenido en el registro «persona» identificado. El objetivo debe escribir los nuevos datos dentro del registro "persona".
changePersonIdentifier	Para cambiar el <i>sourcedId</i> del registro 'persona'. La finalización de esta operación se traducirá en acciones posteriores que utilizan el <i>sourcedId</i> original, registrando un identificador de estatus desconocido.

Tabla 42. Métodos de la interfaz *PersonManager* de la especificación *IMS ES*

- **Interfaz GroupManager:** Es responsable de manipular la información sobre los grupos, apoya la manipulación de un solo objeto grupo y cuenta con los siguientes métodos:

Métodos	Descripción
createGroup	Para solicitar la creación de un registro «grupo» en el sistema de destino y la fuente es responsable de la asignación del identificador único.
deletePerson	Para solicitar la supresión de un registro «grupo». El registro «grupo» es eliminado y todas las relaciones de sus asociados.
deleteGroupRelationship	Para solicitar la supresión de una relación a un registro de «grupo». Este no elimina el correspondiente registro de «grupo».
readGroup	Para leer todo el contenido en el registro «grupo» identificado. El objetivo debe devolver todos los datos del registro «grupo» identificado.
updateGroup	Para escribir nuevos contenidos dentro del registro «grupo» identificado. El objetivo debe escribir los nuevos datos dentro del registro "grupo. Esta

CAPÍTULO 2. DESCRIPCIÓN DE LA SOLUCIÓN PROPUESTA

	operación es un aditivo.
replaceGroup	Para reemplazar el contenido en el registro "grupo" identificado. El objetivo debe escribir los nuevos datos dentro del registro "grupo". Se trata de una escritura más destructiva de toda la información original.
changeGroupIdentifier	Para cambiar el <i>sourcedId</i> del registro 'grupo'. La finalización de esta operación se traducirá en acciones posteriores que utilizan <i>sourcedId</i> original, registrando un identificador de estatus desconocido.

Tabla 43. Métodos de la interfaz *GroupManager* de la especificación *IMS ES*

- **Interfaz MembershipManager:** Es responsable de la manipulación de información sobre la composición de los grupos de personas o grupos, que apoya la manipulación de un solo objeto de miembros y cuenta con los siguientes métodos u operaciones:

Métodos	Descripción
createMembership	Para solicitar la creación de un registro «miembro» en el sistema de destino y la fuente es responsable de la asignación del identificador único.
deleteMembership	Para solicitar la supresión de un registro «miembro». El registro «grupo» es eliminado y todas las relaciones de sus asociados.
readMembership	Para leer todo el contenido en el registro «miembro» identificado. El objetivo debe devolver todos los datos que ha identificado al registro «miembro».
updateMembership	Para escribir nuevos contenidos dentro del registro «miembro» identificado. El objetivo debe escribir los nuevos datos dentro del registro «miembro». Esta operación es un aditivo.
replaceMembership	Para reemplazar el contenido en el registro "grupo" identificado. El objetivo debe escribir los nuevos datos dentro del registro "grupo". Se trata de una escritura más destructiva de toda la información original.
changeMembershipId entfier	Para cambiar el <i>sourcedId</i> del registro 'grupo'. La finalización de esta operación se traducirá en acciones posteriores que utilizan el <i>sourcedId</i> original, registrando un identificador de estatus desconocido.

Tabla 44. Métodos de la interfaz *MembershipManager* de la especificación *IMS ES*

2.3.4. Descripción del estándar *IMS Learner Information Package (IMS LIP)*

Como se ha comentado anteriormente, LIP está basado en un modelo de datos que describe las características de un estudiante. Se identificará en este apartado las fuentes de datos necesarias para satisfacer la especificación LIP según lo planteado por [54].

Dado el volumen de datos que debe manipularse para la especificación LIP, se abordarán por separado cada una de las once estructuras de datos que forman el núcleo de la información de un estudiante y proporcionando para cada uno de los objetos una porción de XML a modo de ejemplo. Los objetos que forman el núcleo son:

Learnerinformation (Información del aprendiz/ principiante)

Estructura de datos raíz de las otras clases. La información de control que describe este tipo de información en si misma está contenida en '*contentype*'.

Un *contentype* de ejemplo podría ser el siguiente [54]

```
<contentype>
  <referential>
    <sourcedid>
      <source>ims_lip_ejemplo_uvuci</source>
      <id>6002</id>
    </sourcedid>
  </referential>
</contentype>
```

La referencia al registro de este estudiante viene identificada por *sourceid*. En nuestro caso *ims_ejemplo_uvuci:6002*. Deberá ser usado para todas las referencias a este estudiante.

En la práctica esto supone garantizar que todas las altas de un estudiante se realizan en gestión académica, para asegurar la correspondencia de su IDP.

Identification (Identificación)

Estructura de datos referente al individuo. Esto incluye datos como: el nombre, la dirección, información de contacto, demográfica y de tutores.

Esta información es puramente administrativa, para identificar y localizar al estudiante. Un registro de este tipo está formado por la siguiente estructura de información [54]:

```
<identification>
  <formname>
    <typename>
      <tysource sourcetype="imsdefault"/>
      <tyvalue>Preferred</tyvalue>
    </typename>
    <comment>texto formateado</comment>
    <text>García Manzano, Pepe</text>
  </formname>
  <name>
    <partname>
      <typename>
        <tysource sourcetype="imsdefault"/>
        <tyvalue>First</tyvalue>
      </typename>
      <text>Pepito</text>
    </partname>
    <partname>
      <typename>
```

```
<tysource sourcetype="imsdefault"/>
  <tyvalue>Last</tyvalue>
</typename>
<text>García</text>
</partname>
<partname>
  <typename>
    <tysource sourcetype="proprietary"/>
    <tyvalue>Segundo apellido</tyvalue>
  </typename>
  <text>Valdemoro</text>
</partname>
</name>
<address>
  <typename>
    <tysource sourcetype="imsdefault"/>
    <tyvalue>Personal</tyvalue>
  </typename>
  <street>
    <nonfieldedstreetaddress>Av. Camilo Cienfuegos</nonfieldedstreetaddress>
    <streetnumber>6</streetnumber>
    <streetname>República Argentina</streetname>
    <streettype>Avenida</streettype>
    <aptnumber>5</aptnumber>
    <aptnumsuffix>1</aptnumsuffix>
  </street>
  <city>Habana</city>
  <region>Habana</region>
  <country>Cuba</country>
  <postcode>08023</postcode>
```

```
</address>
<contactinfo>
  <typename>
    <tysource sourcetype="imsdefault"/>
    <tyvalue>Personal</tyvalue>
  </typename>
  <email>pepito@hotmail.com</email>
</contactinfo>
<contactinfo>
  <typename>
    <tysource sourcetype="imsdefault"/>
    <tyvalue>Trabajo</tyvalue>
  </typename>
  <telephone>
    <areacode>93</areacode>
    <indnumber>44445566</indnumber>
  </telephone>
</contactinfo>
<demographics>
  <typename>
    <tysource sourcetype="imsdefault"/>
    <tyvalue>Adult</tyvalue>
  </typename>
  <gender gender="M"/>
  <date>
    <typename>
      <tysource sourcetype="imsdefault"/>
      <tyvalue>Birth</tyvalue>
    </typename>
    <datetime>1901:04:01</datetime>
```

```
</date>
<placeofbirth>Masnou, Barcelona</placeofbirth>
<uid>464765756Q</uid>
</demographics>
<demographics>
  <representation>
    <typename>
      <tysource sourcetype="imsdefault"/>
      <tyvalue>Photo</tyvalue>
    </typename>
    <description>
      <full>
        <media mediamode="Image" mimetype="image/gif" contentreftype="uri">
          images/foto_de_pepe.gif</media>
        </full>
      </description>
    </representation>
  </demographics>
</identification>
```

No se dispone de información acerca del representante o tutor del estudiante. El objeto agent, por tanto, se omite en el anterior ejemplo.

Accessibility (Accesibilidad)

Información acerca del estudiante referida al conjunto de preferencias de este en aspectos técnicos, físicos, de aprendizaje, capacidades en el lenguaje y adicionalmente tipos de disminución que pueda padecer.

CAPÍTULO 2. DESCRIPCIÓN DE LA SOLUCIÓN PROPUESTA

No se dispone de elementos parametrización más allá de las habilidades del estudiante en el uso de un lenguaje. El siguiente ejemplo contiene la información IMS LIP suficiente para definir las habilidades de un estudiante en el idioma inglés [54].

```
<accessibility>
  <language>
    <typename>
      <tysource sourcetype="imsdefault"/>
      <tyvalue>English</tyvalue>
    </typename>
    <proficiency profmode="OralComp">Good</proficiency>
    <proficiency profmode="OralSpeak">Good</proficiency>
    <proficiency profmode="Read">Excelent</proficiency>
    <proficiency profmode="Write">Poor</proficiency>
  </language>
</accessibility>
```

La especificación contempla además la posibilidad de recoger un conjunto de preferencias acerca del tipo de entrada para el usuario y otros aspectos parametrizables de la presentación del material, uso de fuentes grandes, por ejemplo.

Competency (Competencia)

Información consistente en habilidades que el estudiante ha adquirido. Estas habilidades pueden estar asociadas a aprendizaje formal o informal o histórico de trabajos descritos en ‘*activity*’ o a reconocimiento formal, descrito en ‘*qcl*’.

Goal (Objetivo)

Información que describe objetivos y aspiraciones personales. Se incluye además información para monitorizar el grado de consecución de los objetivos. Las metas adquiridas o adquiribles por un alumno

CAPÍTULO 2. DESCRIPCIÓN DE LA SOLUCIÓN PROPUESTA

dependen en gran medida de los intereses o de las competencias (*competency*) que le empujaron a cursar unos estudios determinados.

En *goal* se engloba todas las aspiraciones del estudiante que no se transforman necesariamente en título, certificación o entregable, pero que se han conseguido.

Qcl (Títulos, Certificaciones y Licencias)

Esta información consiste en las calificaciones, certificaciones y licencias otorgadas al estudiante. Por ejemplo entregables reconocidos formalmente generados a lo largo de su historia académica.

Cabe señalar que en una exportación correcta el origen de datos de este objeto no solo es gestión académica sino también todas las titulaciones existentes en terceros formalmente reconocidas. Estas deberán estar asociadas mediante un objeto relationship a la descomposición descendente de actividades de esta calificación, certificación o licencia.

La siguiente fracción de XML muestra una posible representación de una diplomatura ficticia según el estándar [54].

```
<qcl>
  <typename>
    <tysource sourcetype="imsdefault"/>
    <tyvalue>Diplomatura</tyvalue>
  </typename>
  <contenttype>
    <referential>
      <indexid>qcl_01</indexid>
    </referential>
  </contenttype>
  <title>Diploma en multimedia</title>
  <organization>
```

```
<typename>
  <tysource sourcetype="imsdefault"/>
  <tyvalue>Universidad</tyvalue>
</typename>
<description>
  <short>Universidad de la Ciencias Informáticas</short>
</description>
</organization>
<date>
  <typename>
    <tysource sourcetype="imsdefault"/>
    <tyvalue>Award</tyvalue>
  </typename>
  <datetime>1998:011</datetime>
</date>
</qcl>
```

Activity (Actividad)

Información referente a educación / formación registros y entregables de trabajos y/o servicios de voluntariado o no. Incluyendo información registrada acerca de cursos cursados y sus respectivas calificaciones.

El uso de este objeto es indicado para aquellos cursos y elementos de formación que existan en terceros cuya titulación no sea formalmente reconocida por una organización. Asimismo cualquier labor llevada a cabo en el contexto de la universidad que tenga asociados de alguna forma responsabilidades y logros es susceptible de formar parte del objeto activity.

La siguiente fracción de XML muestra una posible representación de un curso ficticio según el estándar [54].

```
<activity>
  <typename>
    <tysource sourcetype="imsdefault"/>
    <tyvalue>Education</tyvalue>
  </typename>

  <contenttype>
    <referential>
      <indexid>activity_01</indexid>
    </referential>
  </contenttype>

  <units>
    <unitsfield>
      <fieldlabel>
        <typename>
          <tyvalue>CreditsNumber</tyvalue>
        </typename>
      </fieldlabel>
      <fielddata>10</fielddata>
    </unitsfield>
  </units>

  <definition>
    <typename>
      <tysource sourcetype="imsdefault"/>
      <tyvalue>DegreeCourse</tyvalue>
    </typename>

    <definition>
```



```
<contenttype>
  <referential>
    <indexid>Year1</indexid>
  </referential>
</contenttype>
<definition>
  <typename>
    <tyvalue>Modulo</tyvalue>
  </typename>
  <contenttype>
    <referential>
      <indexid>Electronics_01</indexid>
    </referential>
  </contenttype>
  <definitionfield>
    <fieldlabel>
      <typename>
        <tyvalue>Lecture1</tyvalue>
      </typename>
    </fieldlabel>
    <fielddata>BooleanLogic</fielddata>
  </definitionfield>

  <definitionfield>
    <fieldlabel>
      <typename>
        <tyvalue>Lecture2</tyvalue>
      </typename>
    </fieldlabel>
    <fielddata>Transistors</fielddata>
  </definitionfield>
</definition>
```

```
</definitionfield>
</definition>
</definition>
</definition>
</activity>
```

Esta es la estructura para un curso de electrónica. Para simplificar este consiste en un único módulo que se descompone de manera jerárquica en una ristra de lecturas.

La especificación permite incluir entregables que se consideren relevantes dentro de las actividades descritas.

Relationship (Relación)

Es la estructura contenedora para relaciones entre las otras estructuras del núcleo. Un ejemplo de relación podría ser la existente entre *qcl* y la organización que otorga la titulación. Esta estructura permite la realización de complejas estructuras entre diversas estructuras del núcleo.

A continuación se muestra la relación establecida entre una actividad y su *qcl* de reconocimiento asociado. Podría ser interesante asociar también competencias (competency) y metas (goals) adquiridos durante el desarrollo de cada una de las actividades [54].

```
<relationship>
  <typename>
    <tyvalue>qcl</tyvalue>
  </typename>
  <tuple>
    <tuplesource>
      <indexid>qcl_01</indexid>
    </tuplesource>
```

```
<tuplerelation>
  <typename>
    <tyvalue>resultsfrom</tyvalue>
  </typename>
</tuplerelation>
<tupledest>
  <indexid>activity_01</indexid>
</tupledest>
</tuple>
</relationship>
```

Transcript (Transcripción)

Registro sumario del rendimiento académico de un individuo. Típicamente proporcionado por el organismo responsable de evaluar al individuo.

El estándar permite vincular un documento externo, documento este que suponemos contenedor del sumario emitido. La siguiente porción vincula un documento indicando la fecha en que ha sido emitido [54].

```
<transcript>
  <typename>
    <tyvalue></tyvalue>
  </typename>
  <contenttype>
    <referential>
      <indexid>transcript_01</indexid>
    </referential>
  </contenttype>
  <exrefrecord>
    <recformat>MSWord98</recformat>
```

```
<recdata uri="myreferences/reference.doc"></recdata>
<date>
  <datetime>1999:10:03</datetime>
</date>
</exrefrecord>
</transcript>
```

Securitykey (Llave de seguridad)

Esta información se usa para definiciones de passwords, certificados, PINs y claves de autenticación. Estas claves se usan para interactuar con el estudiante.

El aspecto del objeto securitykey para almacenar una clase es el siguiente [54]:

```
<securitykey>
  <typename>
    <tyvalue>Password</tyvalue>
  </typename>
  <contenttype>
    <referential>
      <indexid>securitykey_1</indexid>
    </referential>
  </contenttype>
  <keyfields>
    <fieldlabel>
      <typename>
        <tyvalue>Password</tyvalue>
      </typename>
    </fieldlabel>
    <fielddata>pmorcilla34</fielddata>
  </keyfields>
```

</securitykey>

Interest (Interés)

Información sobre el estudiante que se refiere a actividades no lectivas u ociosas, como hobbies. Estas actividades pueden tener reconocimiento formal, descrito en el 'qcl' asociado. No se contempla guardar en ningún momento información relativa a este registro.

Affiliation (Afiliación)

Información descriptiva de afiliaciones del estudiante. Se permite la creación de asociaciones entre estudiantes, pero su funcionamiento es autónomo. No depende de campus ni de gestión administrativa.

2.3.5. Descripción de los métodos de IMS Digital Repositories Interoperability (IMS DRI)

IMS DRI, como se abordó en el capítulo anterior, es el estándar utilizado para la interoperabilidad entre herramientas que realizan las funciones básicas de un repositorio, que son Buscar/Exponer, Colectar/Exponer, Enviar/Almacenar, Pedir/Entregar y Alertar/Exponer y para ellas recomienda varios estándares.

Para la UVUCI se propone que se utilicen solo tres de estas funciones (Buscar/Exponer, Enviar/Almacenar y Pedir/Entregar), homológamente a como fuera definido y propuesto en el FI, pues son las básicas para establecer una comunicación con otras aplicaciones, aunque no serán aplicados con todos los estándares recomendados, por las facilidades que proveen otros estándares. Estas funciones serán implementadas usando servicios web, los que a su vez formarán parte de una arquitectura SOA.

Para el caso de la función **Enviar/Almacenar** se tendrá en cuenta el estándar que propone IMS- DRI, es decir, el uso de mensajes SOAP para el envío de contenidos entre aplicaciones. En cambio para los métodos **Pedir/Entregar** y **Buscar/Exponer** se realizará con el estándar SQLI y no con OAI como lo recomienda IMS DRI, pues OAI sólo es posible utilizarlo para repositorios que utilizan DublinCore como

CAPÍTULO 2. DESCRIPCIÓN DE LA SOLUCIÓN PROPUESTA

esquema de metadatos, principalmente en bibliotecas digitales, Al contrario, SQL es neutral en términos de lenguaje de consultas, es decir, no especifica cómo van a estar descritos los OA, ni la estructura que tendrá la consulta.

2.3.6. Descripción de los servicios ANSI / NISO Z39.50

Como fuera mencionado con anterioridad ANSI / NISO Z39.50 normaliza el procedimiento para que dos ordenadores se comuniquen cuando se desea recuperar información recomendado a utilizar en la búsqueda y recuperación de información en la BV, para ello se deben aplicar los 11 facilidades mínimos que deben contemplarse para negociar el inicio de una sesión, realizar una búsqueda en una base de datos, crear un conjunto de resultados en dicha base que se ajusten a la búsqueda y obtener uno o varios de los registros del conjunto. Cada una de estas facilidades contiene uno o más servicios.

A continuación se especifican algunos parámetros de los mismos según [55]:

- **Initialization Facility**

La facilidad de inicialización consiste en la inicialización de un único servicio, denominado *Init*, que permite al cliente negociar una Z-asociación (funcionalidad soportada, juego de caracteres, idioma, etc.). En la solicitud de *Init*, el cliente propone los valores de los parámetros de inicialización. En la respuesta *Init*, el servidor responde con los valores para la inicialización de los parámetros.

Si el servidor responde afirmativamente (Result = 'aceptar'), la Z-asociación se establece. Si el el cliente no desea aceptar los valores de la respuesta del servidor, se podrá poner fin a la Z asociación, a través del Cierre de servicios (y por consiguiente intento de inicializar de nuevo). Si el servidor responde negativamente, el cliente puede intentar inicializar de nuevo.

Como parámetros obligatorios se proponen los siguientes:

Parámetros	Descripción
Versión	Tanto el cliente y el servidor indican todas las versiones que soportan. La máxima versión común es seleccionada para su uso, y se dice que está "en vigor", por la Z-

CAPÍTULO 2. DESCRIPCIÓN DE LA SOLUCIÓN PROPUESTA

	asociación. Si no existen versiones en común, el servidor debe indicar "rechazar" para el parámetro de resultado.
Options	Para cada una de las capacidades en el cuadro siguiente, el cliente propone ya sea 'on' o 'off' (que significa "en efecto" o "no está en vigor», respectivamente) y el servidor responde correspondientemente a cada uno. La respuesta determina si la capacidad está en vigor.
Preferred-message-size	La respuesta de inicio contiene el <i>Preferred-message-size</i> (tamaño del mensaje de preferencia) y <i>Exceptional-record-size</i> (tamaño de registro excepcional) que el servidor va a utilizar, estos pueden ser diferente a partir de (y sobre) los valores propuestos por el cliente. Tanto para la solicitud y la respuesta, <i>Preferred-message-size</i> debe ser menor o igual a <i>Exceptional-record-size</i> . Ambas son especificadas en Byte.
Exceptional-record-size	Es de tamaño significativo durante una operación actual, y sólo en el caso especial cuando un único registro excepcionalmente grande (es decir, más grande que <i>Preferred-message-size</i> de tamaño) se solicita en la presente petición.
Result	El servidor indica si acepta o no el Z-asociación especificando un valor de "aceptar" o "rechazar" en el parámetro de resultados. (Si «rechazar» se indica, el cliente puede enviar otra solicitud de inicio.

Tabla 45. *Parámetros del servicio de Initialization Facility de Z39.50*

- **Search Facility**

La facilidad de búsqueda consiste de un solo servicio, Search (Buscar). El servicio de búsqueda permite al cliente consultar las bases de datos de un servidor, crear el conjunto de resultados en el servidor y recibir información sobre dicho conjunto, la eficiencia de los mismos depende de los parámetros de búsqueda, seguidamente se especifican los parámetros obligatorios:

Parámetros	Descripción
Query-type	Identifica los tipos de consulta, es decir, la sintaxis de los parámetros de la consulta. Existen 7 tipos:

CAPÍTULO 2. DESCRIPCIÓN DE LA SOLUCIÓN PROPUESTA

	<p>Tipo-0 puede ser utilizado sólo cuando el cliente y el servidor tienen una prioridad, fuera de la estándar.</p> <p>Tipo-1 es la Notación Polaca Inversa (RPN) de consulta</p> <p>Tipo-2 es el tipo de consulta ISO8777, especificados en la norma ISO 8777.</p> <p>Tipo-100 es el lenguaje común de comandos de consulta</p> <p>Tipo-101 es la ampliación de la consulta de RPN (ERP)</p> <p>Tipo-102 es la clasificación de consulta, que deberán definirse en una versión posterior de esta norma.</p> <p>Tipo-104 se utiliza para identificar las preguntas definidas externamente</p>
Database-names	El cliente indica el conjunto de bases de datos a los que se aplica la consulta.
Result-set-name	Especifica un nombre en el que se aprueba el conjunto de resultados (que se creará por la consulta) para que pueda ser posteriormente referenciada.
Replace-indicator	Para reemplazar el identificador o el Result-set-name.
Search-status	<p>El parámetro de búsqueda de estado, retorna en la respuesta, uno de los dos valores siguientes:</p> <p>El éxito de búsqueda se ha completado satisfactoriamente.</p> <p>El fracaso de búsqueda no se ha completado con éxito.</p>

Tabla 46. *Parámetros del servicio de Search Facility de Z39.50*

- **Retrieval Facility**

La facilidad de Recuperación consta de dos servicios: Present y Segment.

El servicio **Present** permite al cliente solicitar la respuesta correspondiente a la base de datos de registros representado por los registros de un determinado resultado. Este servicio cuenta con los siguientes parámetros:

Parámetros	Descripción
Number-of-records- requested	Número de registro solicitado a partir del Result-set-start-position.

CAPÍTULO 2. DESCRIPCIÓN DE LA SOLUCIÓN PROPUESTA

Result-set-start-position	Resultado de la posición de salida.
Result-set-id	El cliente puede indicar la composición ideal de los registros recuperados

Tabla 47. Parámetros del servicio de Retrieval Facility (Present) de Z39.50

En el servicio **Segment**, si los registros solicitados por una solicitud Present no caben en un solo segmento, y si la segmentación es, en efecto, el servidor devuelve varios segmentos, cada uno de los cuales contiene una parte de los registros.

A continuación se especifican los parámetros de este servicio:

Parámetros	Descripción
Segment-records	Si el nivel 1 de segmentación es, en efecto, el Segment-records (Segmentos de registros) consiste en una secuencia de respuestas de los registros. Si el nivel 2 de segmentación es, en efecto, el Segment-records puede incluir respuesta de los registros así como fragmentos
Number-of-records-returned	Este es el número total de registros y la respuesta a partir de fragmentos incluidos en el segmento.

Tabla 48. Parámetros del servicio de Retrieval Facility (Segment) de Z39.50

- **Result-set-delete Facility**

La facilidad eliminar-cambiar-Resultado consiste de un único servicio, Delete (Eliminar). El servicio Eliminar permite que un cliente pueda solicitar a un servidor especificado que se borre un conjunto de resultados determinado, o todos ellos, creados durante la Z-asociación. El servidor responde por la presentación de información relacionada con el resultado de la operación. El parámetro obligatorio propuesto por este servicio es:

Parámetros	Descripción
Delete-function	El cliente especifica una de las siguientes: Lista borra los conjuntos de resultados especificados.

CAPÍTULO 2. DESCRIPCIÓN DE LA SOLUCIÓN PROPUESTA

	Conjunto-eliminar borra todos los conjuntos de resultados creados actualmente en el servidor durante este Z-asociación.
--	--

Tabla 49. Parámetros del servicio de Result-set-delete Facility de Z39.50

- **Access Control Facility**

La facilidad de Control de Acceso consta de un único servicio, Control de Acceso. El servicio de control de acceso permite a un servidor desafiar un cliente. El desafío podría pertenecer a una operación específica o la Z-asociación. El mecanismo de control de acceso solicitud/respuesta puede ser utilizado para apoyar los desafíos de control de acceso o autenticación, incluidos los desafíos de contraseña, cripto sistemas de clave pública, y algoritmos de de autenticación. Los parámetros propuestos por este servicio son:

Parámetros	Descripción
Security-challenge	Definiciones del formato y el contenido del desafío y la respuesta son objeto de registro. Alternativamente, el contenido de estos dos parámetros puede ser establecida por acuerdo previo entre un par cliente / servidor. Security-challenge es la solicitud del servidor.
Security-challenge-response	Se ocupa de la respuesta del cliente.

Tabla 50. Parámetros del servicio de Access Control Facility de Z39.50

- **Accounting/Resource Control Facility**

La facilidad Control cuenta/recurso consta de tres servicios: el **Resource-control**, invocado por el servidor, ya sea como parte de una operación (de cualquier tipo), o relativas a la Z-asociación, el **Trigger-resource-control**, invocadas por el cliente como parte de una operación activa (de cualquier tipo, excepto de inicio) y el servicio **Resource-report**, invocado por el cliente para iniciar una operación de recursos-informe. A continuación serán explicados cada uno de estos servicios así como sus parámetros obligatorios.

CAPÍTULO 2. DESCRIPCIÓN DE LA SOLUCIÓN PROPUESTA

El servicio *Resource-control* permite al servidor enviar una petición de control de recursos, el cual podría incluir un informe de recursos. En el informe se podría notificar al cliente sobre el consumo de los recursos, si serán superiores a los límites acordados (o límites construido en el servidor), y solicitar el consentimiento del cliente para continuar una operación, a través de la respuesta del control los recursos. El servidor podría, por ejemplo, informar al cliente sobre el estado actual de un conjunto de resultados que se generan en el servidor durante una operación de búsqueda, e indicar la información sobre el progreso de la operación. Contiene los parámetros:

Parámetros	Descripción
Response-required	Este parámetro se puede utilizar para transmitir información sobre la situación actual y la estimación de los recursos de consumo en el servidor.
Continue-flag	Este parámetro sólo es válido cuando la solicitud se refiere a una operación. El cliente indica al servidor si continuar o no con la tramitación de la operación.

Tabla 51. Parámetros del servicio de Accounting/Resource Control (*Resource-control*) de Z39.50

El servicio *Trigger-resource-control* permite al cliente solicitar que se inicie el control de recursos o cancelar la operación. Contiene los parámetros:

Parámetros	Descripción
Requested-action	El cliente indica uno de los siguientes: Resource-report (informe de recursos): Emite una solicitud de control de recursos y cambia las respuestas requeridas a “off”. Resource- control (control de recursos): Emite una solicitud de control de recursos y cambia las respuestas requeridas a “on”. Cancel : Termina la operación

Tabla 52. Parámetros de Accounting/Resource Control (*Trigger-resource-control*) de Z39.50

Resource-report es un servicio que permite al cliente solicitar un informe de recursos tanto de una operación, como de una sesión completa. O una Z-association). Contiene los parámetros:

CAPÍTULO 2. DESCRIPCIÓN DE LA SOLUCIÓN PROPUESTA

Parámetros	Descripción
Resource-report-status	<p>El servidor provee uno de los siguientes valores de status:</p> <p>Success: Un informe de los recursos se incluye (y en el formato preferido si el parámetro Preferred-resource-report-format fue incluido en la solicitud).</p> <p>Partial: Un informe de los recursos se incluye, pero no en el formato preferido (sólo se aplica si el parámetro Preferred-resource-report-format se incluyó en la solicitud).</p> <p>Failure-1: Servidor incapaz de proveer un reporte de recurso</p> <p>Failure-2: Operación terminada por el servidor debido a restricciones de recursos limitaciones.</p> <p>Failure-3: Control de acceso fallido.</p> <p>Failure-4: Fallo no especificado.</p> <p>Failure-5: No se conoce de alguna operación con el id especificado.</p> <p>Failure-6: Hay una operación con el id especificado.</p>

Tabla 53. Parámetros de Accounting/Resource Control Facility (Resource-report) de Z39.50

- **Sort Facility**

Ordenar (Sort), es una facilidad que consta de dos servicios:

Ordenar Servicio (sort) permite a un cliente solicitar un conjunto de resultados ordenados del servidor (o unir múltiples conjuntos de resultados y luego ordenarlos). El cliente especifica una secuencia de elementos ordenados. El conjunto de resultados será ordenado atendiendo a la secuencia especificada y las solicitudes subsecuentes de posiciones en contraposición con los resultados que serán construidos por el servidor para aplicarlos a los resultados de acuerdo a como los ordena. Este servicio contiene cuatro parámetros obligatorios: *Input-result-sets*, *Sorted-result-set*, *Sort-sequence* y *Sort-status*

CAPÍTULO 2. DESCRIPCIÓN DE LA SOLUCIÓN PROPUESTA

Detección de duplicados (Duplicate-detection) permite al cliente solicitar que el servidor de analice uno o más conjuntos de resultados en términos de posibles duplicados y construir un nuevo conjunto de resultados de acuerdo con los criterios especificados por el cliente para detectar, retener, agrupar y ordenar los registros incluidos duplicados. Contiene los parámetros: *Input Result Set Id, Output Result Set Name, Retention Criterion, Status*.

- **Browse Facility**

Facilidad de Navegar, consiste en un único servicio, el Scan (exploración).

El servicio de rastreo o exploración se utiliza para escanear una lista ordenada de términos (términos sujeto, nombres, títulos, etc.) El orden de la lista de términos es definido por el servidor. El cliente especifica una lista de términos a explorar así como un término inicial (de forma implícita, mediante la especificación de una combinación atributo / término combinación de plazo y un id de base de datos), el tamaño de las medidas de exploración, y el número deseado de entradas además de la posición del elemento de inicio en la respuesta. Los parámetros propuestos de este servicio son:

Parámetros	Descripción
Database Names	Identifica un conjunto de bases de datos a los que el término lista (especificado por Term-list-and-start-point) pertenece.
Term-list-and-start-point	El cliente suministra una lista de atributos y términos. La lista contiene atributos que indican cual términos escanear. El término es calificado por esos atributos, indicando donde comienza la exploración, esta podría ser una entrada presumida en la lista de términos. Si no hay entradas que coincidan, la primera entrada con el valor más alto será el punto de partida.
Number-of-entries	El cliente indica el número propuesto de entradas a ser devueltas. El servidor indica el número de entradas devueltas. Si el número real es menor que el número propuesto, la razón, indicado el status de la exploración.

Tabla 54. Parámetros del servicio de Browse Facility de Z39.50

- **Extended Services Facility**

CAPÍTULO 2. DESCRIPCIÓN DE LA SOLUCIÓN PROPUESTA

La facilidad de Servicios Extendidos consta de un único servicio, Extended-services. Esta facilidad permite el acceso a servicios ajenos al protocolo que pueden perdurar una vez termine la Z-asociación como búsquedas periódicas, conservar conjuntos de resultados, etc.

El servicio Extended-services permite a un cliente crear, modificar o borrar un paquete de tareas en el servidor. El servidor mantiene paquetes de tareas en una base de datos especial. Los parámetros propuestos por este servicio son:

Parámetros	Descripción
Function	El cliente especifica Crear, Suprimir o Modificar. Si la función es Crear, el servidor crea un paquete de tareas y le asigna el nombre especificado por el parámetro <i>nombre_paquete</i> , si este es provisto.
Package-type	El paquete tipo identifica el servicio extendido solicitado. La ampliación de los servicios definidos por esta norma son: Guardar un conjunto de resultados para su uso posterior Guardar una consulta para su uso posterior Definir un calendario periódico de búsqueda Ordene un elemento. Actualización de una base de datos Crear una especificación de exportación. Invoque una especificación de exportación creada previamente
Task-specific-parameters	Estos son los parámetros adicionales, que se define por el servicio extendido específicos.
Wait-action	El cliente indica si el servidor debe (o puede) el paquete de tareas en la respuesta ES. Este parámetro tiene cuatro posibles valores: Wait: El servidor debe realizar la tarea antes de emitir la respuesta ES (a menos que aborte la operación). Si no se está dispuesto a realizar la tarea antes de emitir la respuesta, se debe denegar la solicitud de responder con un estado de 'fracaso' y un diagnóstico adecuado. Si el servidor acepta la petición, este incluye el parámetro Task-package en la respuesta.

CAPÍTULO 2. DESCRIPCIÓN DE LA SOLUCIÓN PROPUESTA

	<p>Wait-if-possible: El cliente pide que, si es posible, el servidor de realice la tarea antes de emitir la respuesta ES e incluya el paquete de tareas en la respuesta. Si no fuera posible, el servidor debe proceder como si el valor fuese “no esperar”.</p> <p>Do-not-wait: El cliente no solicita que el servidor intente realizar la tarea antes de emitir la respuesta ES. Sin embargo, si el servidor no realiza la tarea antes de emitir la respuesta, entonces la respuesta puede incluir el paquete de tareas.</p> <p>Do-not-send-task-package: El servidor puede realizar la tarea cuando se elige, pero no es para incluir el paquete de tareas en la respuesta bajo cualquier circunstancia.</p>
Operation-status	<p>Este es el estado de operación. Es uno de los siguientes:</p> <p>Done: La solicitud es aceptada, la tarea es completa y los resultados son incluidos en el paquete de tareas.</p> <p>Accepted: La solicitud es aceptada y la tarea es puesta en cola para procesamiento, o está en proceso.</p> <p>Failure: La solicitud es denegada. Uno o más diagnósticos son suministrados (en el parámetro <i>Operation-diagnostics</i>).</p>

Tabla 55. Parámetros del servicio de Extended Services Facility de Z39.50

- **Explain Facility**

La facilidad Explain ofrece detalles del servidor como: bases de datos disponibles, índices, servicios disponibles, etc. con idea de que se puedan desarrollar clientes que se auto configuren en función de los servidores que encuentren.

- **Termination Facility**

La facilidad Termination cuenta con un solo servicio, Close. Este servicio permite al cliente o al servidor interrumpir las operaciones activas e iniciar el cierre de la Z-asociación. El parámetro obligatorio propuesto para este servicio es:

Parámetros	Descripción
------------	-------------

CAPÍTULO 2. DESCRIPCIÓN DE LA SOLUCIÓN PROPUESTA

Close-reason	Este parámetro indica la razón por la cual el cliente o el servidor cierran la Z-associación. Su valores son: <i>Terminado, Cierre, Problema en el sistema, Límites de coste, Recursos, violación de seguridad, Error de protocolo, Falta de actividad, Sin especificar, Respuesta a la solicitud Cerrar.</i>
--------------	---

Tabla 56. Parámetros del servicio de Extended Termination Facility de Z39.50

2.3.7. Estándar de Seguridad Web Service Security (WS-Security)

Como se vio en el capítulo 1, WS Security es el estándar propuesto para garantizar el envío seguro de datos en la UVUCI. El mismo proporciona seguridad al mensaje SOAP, como se puede ver en la Figura 10, WSS, incorpora en la cabecera aspectos de seguridad como: encriptación o codificación con XML Encryption, firmado de mensaje con XML Digital Signature y como mecanismo para asociar a las fichas de seguridad con el contenido de los mensajes el estándar SAML.

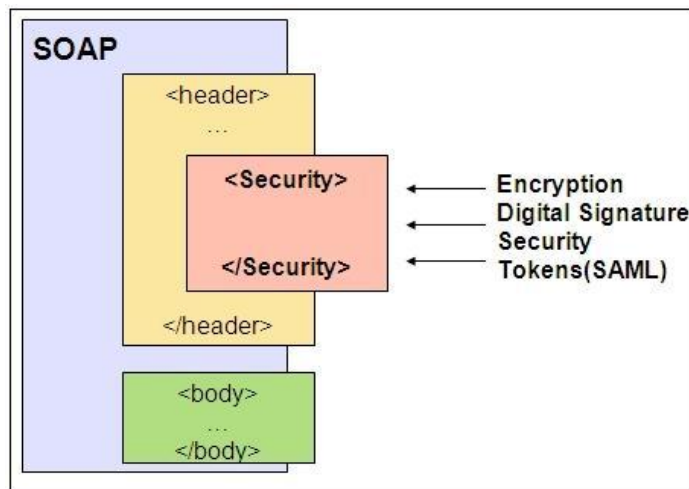


Figura 10. Estructura de un mensaje SOAP con WS Security

A continuación se explicará a través de un ejemplo en XML (Ver figura 11), cómo se incorpora la seguridad en un mensaje SOAP y algunos de los aspectos a incluir utilizando los estándares XML Encryption, XML Digital Signature y SAML.

CAPÍTULO 2. DESCRIPCIÓN DE LA SOLUCIÓN PROPUESTA

```
...
(004) <S: header> ...
(005)     <wsse: Security xmlns:wsse="http://schemas.xmlsoap.org/ws/2002/04/secext">
(006)         <saml: Assertion
                MajorVersion="1" MinorVersion="0"
                AssertionID="SecurityToken-ef912422"
                Issuer="jselman"
                IssueInstant="2005-05-14T16:47:05.6228146-07:00"
                xmlns:saml="urn:oasis:names:tc:SAML:1.0:assertion">
(007)             <saml: AuthenticationStatement
                    AuthenticationMethod="password"
                    AuthenticationInstant="2005-05-07T22:02:00Z">
(008)                 <saml: Subject>
(009)                     <saml: NameIdentifier
                            SecurityDomain="jselman.com"
                            Name="José Miguel" />
(010)                 </saml: Subject>
(011)             </saml: AuthenticationStatement> ...
(012)         </saml: Assertion>
(013)         <xenc: EncryptedData
                xmlns="http://www.w3.org/2001/04/xmlenc#"
                Type="http://www.w3.org/2001/04/xmlenc#Element"/>
(014)             <EncryptionMethod
                    Algorithm="http://www.w3.org/2001/04/xmlenc#tripleDES-cbc"/>
(015)                 <ds:KeyInfo
                        xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
                        <ds:KeyName> José Miguel </ds:KeyName>
(016)                 </ds:KeyInfo>
(017)                 <CipherData>
(018)                     <CipherValue>DEADBEEF</CipherValue>
(019)                 </CipherData>
(020)             </xenc: EncryptedData>
(021)         <ds:Signature>
(022)             <ds:SignedInfo>
(023)                 <ds:CanonicalizationMethod
                        Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#" />
(024)                 <ds:SignatureMethod
                        Algorithm="http://www.w3.org/2000/09/xmldsig#hmac-sha1" />
(025)                 <ds:Reference URI="#MsgBody">
(026)                     <ds:DigestMethod
                            Algorithm="http://www.w3.org/2000/09/xmldsig#sha1" />
(027)                     <ds:DigestValue>LyLsF0Pi4wPU...</ds:DigestValue>
(028)                 </ds:Reference>
(029)             </ds:SignedInfo>
(030)             <ds:SignatureValue>Djbchm5gK...</ds:SignatureValue>
(031)                 <ds:KeyInfo>
(032)                     <wsse:SecurityTokenReference
                            <wsse:Reference URI="#MvID" />
(033)                     </wsse:SecurityTokenReference>
(034)                 </ds:KeyInfo>
(035)             </ds:Signature>
(036)         </wsse: Security>
(037)     </S: header>
(038) ...
(039)
```

CAPÍTULO 2. DESCRIPCIÓN DE LA SOLUCIÓN PROPUESTA

Figura 11. Ejemplo de un fragmento de código del estándar WS Security [56]

- Línea (005). Especifica una cabecera <Security> que contiene información de seguridad para el receptor deseado. Este elemento continúa hasta la línea (038).
- Líneas (006) a (012). Estas líneas especifican un token de seguridad que está asociado con el mensaje basado en SAML. En este caso, el mismo define la menor y mayor versión (MajorVersion="1" MinorVersion="0"), la emisora (Issuer="jselman"), la emisora instantánea (IssueInstant="2005-05-14T16:47:05.6228146-07:00") y el espacio de nombres que va a ser utilizado en SAML (xmlns:saml="urn:oasis:names:tc:SAML:1.0:assertion").
- Línea (007) hasta la (011). En estas se encuentran las declaraciones de autenticación (saml:AuthenticationStatement), especificando como atributos el método de autenticación (AuthenticationMethod="password") y la autenticación instantánea (AuthenticationInstant="2005-05-07T22:02:00Z">).
- Línea (008). Especifica el tema de la declaración instantánea de SAML. En este ejemplo el tema es de identificador del nombre, en el que se especifica el dominio de seguridad (SecurityDomain="jselman.com") y el nombre (Name="José Miguel").
- Línea (013). El tipo de cifrado de datos puede ser representado como un atributo de valor para ayudar en la descodificación y posterior procesamiento. En este caso, el cifrado de datos es un "elemento". Otras alternativas incluyen "contenido" de un elemento, o una secuencia de octeto externos que también pueden ser identificados a través de los atributos MimeType y Encoding.
- Línea (014). Este (3DES CTF) es un sistema de cifrado de clave simétrica.
- Línea (016). La clave simétrica tiene un nombre " José Miguel".
- Línea (018) hasta (020).El elemento CipherData contiene una CipherValue que es la base de secuencia codificada de un octeto. Alternativamente, podría contener una CipherReference que es un URI de referencia junto con transformaciones necesarias para obtener los datos como una secuencia de octeto.
- Líneas (022) a (037). Especifican una firma digital. Esta firma asegura la integridad de los elementos firmados (que los mismos no están modificados). La firma respeta la especificación XML Signature. En este ejemplo, la firma se basa en una clave generada a partir del password del usuario.

CAPÍTULO 2. DESCRIPCIÓN DE LA SOLUCIÓN PROPUESTA

- Líneas (023) a (030) describe la firma digital. La línea (024) especifica cómo canonizar (normalizar) los datos que están siendo firmados.
- Líneas (026) a (029) selecciona los elementos que están firmados. Especialmente, la línea (026) que está firmado el elemento <S: Body>. En este ejemplo, sólo está firmado el cuerpo del mensaje. En estas líneas el elemento <DigestValue> lleva el valor del resumen del recurso referenciado.
- Línea (031) especifica el valor de la firma del formato normalizado de los datos que están siendo firmados como se define en la especificación de XML Signature.
- Líneas (032) a (036) provee un indicio de dónde encontrar el token de seguridad asociado con esta firma. El elemento <KeyInfo> indica la llave que se usará para validar la firma y las posibles formas de identificación incluyendo certificados, nombres de llaves, acuerdos de llaves y algoritmos. Especialmente, las líneas (033) y (034) indican que el token de seguridad se puede encontrar (se puede extraer) del URL especificado.

2.4. Niveles de madurez en la UVUCI

Como fuera enunciado en secciones anteriores, el modelo arquitectónico propuesto tiene por basamento al FI de manera que puede tomarse como una extensión refinada de dicho framework. Tomando como punto de partida este planteamiento, ha de destacarse que la propuesta de solución brindada considera y adoptar los mismos niveles de madurez (Nivel 1 (Punto a Punto), Nivel 2 (UDDI/ESB) y Nivel 3 (BPM)) propuestos por el FI siguiendo una arquitectura SOA, para un mejor entendimiento consultar [4].

Se propone como nivel de madurez de SOA a alcanzar en las primeras versiones de la UVUCI, el nivel 2 fase 1 (Ver Figura 12), el cual permite gestionar mediante una UDDI los servicios web de las aplicaciones haciendo uso de un catálogo de servicios. Primeramente debe alcanzarse el nivel 1 (Punto a Punto), el cual implementará como servicios web las OSIDs authentication, authorization y logging así como los métodos correspondientes a SQI, IMS-DRI, IMS-ES e IMS-LIP. Para ello se propone XML, el protocolo SOAP y para que los servicios permitan distribuir los detalles en un lenguaje neutro, se utiliza WSDL.

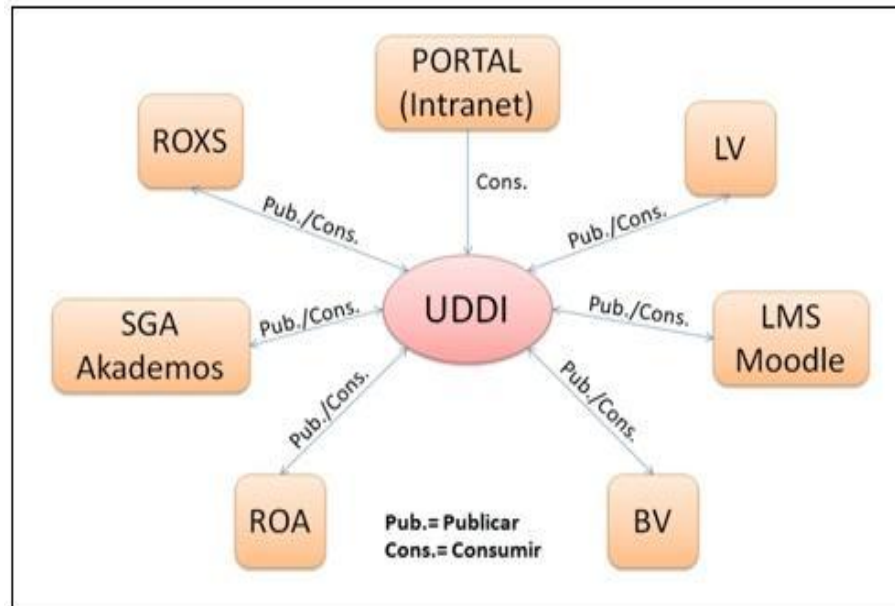


Figura 12. Nivel 2 de la UVUCI a través de una UDDI

2.5. Trabajo Futuro

Como propuesta futura para la UVUCI se plantea:

- Identificar todos los servicios que brindará cada uno de los componentes que integran el CV.
- Se exhorta a la inclusión de nuevas OSIDs de OKI para así facilitar la comunicación entre los componentes propuestos y por otro lado poder darle cobertura a otros componentes que se identifiquen en el futuro y se deseen integrar e intercomunicar con los ya existentes en el CV.
- Incluir estándares para que en las aulas virtuales se pueda transmitir las Teleconferencias en Vivo y realizar el Monitoreo por Estaciones de Trabajo.
- Realizar un estudio sobre las tecnologías y plataformas a utilizar en la futura implementación de este modelo.
- Profundizar en la calidad de los servicios de la arquitectura SOA, en cuanto a políticas, seguridad, transacciones y administración.
- Alcanzar los niveles de madurez 4 y 5 de SOA.

2.6. Conclusiones del Capítulo

En este capítulo se explicaron los principales procesos o servicios básicos que tendrán lugar dentro del modelo de universidad virtual propuesto. Se realizó una explicación detallada de la arquitectura que sirve de soporte al CV. Por otro lado se hizo una descripción de los estándares utilizados en la solución así como los métodos empleados por cada uno de estos estándares definiendo para ello los parámetros y valor de retorno de los mismos. Así mismo son presentados los niveles de madurez a alcanzar por la solución teniendo en cuenta los propuestos por el framework integrador para una arquitectura SOA. Por último, se realizó una breve descripción del trabajo futuro de la UV para la UCI.

CONCLUSIONES GENERALES

Se cumplieron los objetivos del trabajo, obteniéndose resultados satisfactorios.

Además:

- Fueron identificados los componentes y herramientas que componen el CV para la UCI.
- Para lograr la interoperabilidad entre los componentes del CV fueron identificados y descritos los estándares de interoperabilidad IMS DRI, IMS ES, IMS LIP, Z39.50, SQI y OKI así como el estándar de seguridad: WS-Security y sus dependencias.
- Se propuso la arquitectura base de un modelo para la implementación de una Universidad Virtual para la UCI.

RECOMENDACIONES

Para dar continuidad al presente trabajo se recomienda:

- Mantener la vigilancia tecnológica sobre la evolución de los estándares utilizados y las mejores propuestas de UVs a nivel mundial en pos de poder identificar nuevos componentes y así poder extender las funcionalidades y servicios del CV.
- Realizar un estudio de otros estándares de interoperabilidad y seguridad para valorar su inclusión en el futuro desarrollo de esta investigación.
- Profundizar en la calidad de los servicios de la arquitectura SOA.
- Implementar el modelo de interoperabilidad entre los componentes de una UV que propone esta investigación, utilizando los componentes existentes en la UCI, requiriéndose para ello que algunos de los componentes propuestos y aún en fase de desarrollo, estén disponibles al menos en sus versiones iniciales, así como, ajustarse al modelo arquitectónico que se presentó.

REFERENCIAS BIBLIOGRÁFICAS

- [1]. Red TTnet. *La formación sin distancia. Estudio realizado por el Grupo de Trabajo de e-Learning de la red TTnet* . 2005.
- [2]. Martí Arias, José. INTRODUCCIÓN Y DESARROLLO DEL E-LEARNING EN EL CENTRO DE ESTUDIO DE ADMINISTRACIÓN PÚBLICA. [En línea] 2004. [Citado el: 3 de marzo de 2009.] http://74.125.47.132/search?q=cache:B_mmqNmYQN4J:www.clad.org.ve/documentos/otros-documentos/marti+desarrollo+de+la+inform%C3%A1tica+%2B+desarrollando+el+e-learning&hl=es&ct=clnk&cd=8&gl=cu..
- [3]. Universidad Virtual UTM. Universidad Virtual de la Maxteca. . [En línea] 2006. [Citado el: 10 de noviembre de 2008.] <http://virtual.utm.mx/>.
- [4]. Cañizares Glez, Roxana. *Framework para la gestión de contenidos educativos*. Ciudad de la Habana: : s.n., 2008.
- [5]. Tamayo Ávila, D. *Herramientas para la reutilización de contenidos educativos* . 2007.
- [6]. BIDIPED. Biblioteca Digital para personas con Discapacidad. Glosario de Términos. Biblioteca Virtual. [En línea] 2007. [Citado el: 11 de noviembre de 2008.] <http://www.oit.or.cr/bidiped/Glosario.html>.
- [7]. Hernández Ferrer, Kenia y Vázquez Mora, Miriolka. *Laboratorio Virtual de Física: Propuestas de nuevas Funcionalidades en la UCI*. Ciudad de la Habana, Cuba : s.n., 2007.
- [8]. La Menza, Mario Alberto. Introducción a la arquitectura orientada a servicio. Introducción . [En línea] 2007. [Citado el: 8 de diciembre de 2008.] http://www.adeideas.com/paquito/archives/174/Arquitectura_orientada_servicio.htm.
- [9]. CSIC. Ranking Web de Universidades del Mundo. Ranking Mundial. [En línea] 2008. [Citado el: 15 de diciembre de 2008.] http://www.webometrics.info/top4000_es.asp?offset=0..
- [10]. Jardines Méndez, J. B. *Educación en red: mucho más que Educación a Distancia. Experiencias de las universidades médicas cubanas*. 2007.
- [11]. Biblioteca Digital. Servicios. [En línea] 2005. [Citado el: 15 de noviembre de 2008.] <http://biblioteca.uci.cu/bives/informacion.htm>.
- [12]. Aula Global. *Observatorio de e-Learning*. 2005.

- [13]. USMPVirtual. Universidad Virtual. . [En línea] 2006. [Citado el: 30 de enero de 2009.] [http://www.usmpvirtual.edu.pe/.](http://www.usmpvirtual.edu.pe/)
- [14]. Lopez Guzmán, Clara. *Los Repositorios de Objetos de Aprendizaje como soporte a un entorno e-learnin. Tesina doctoral.* España, Salamanca : Universidad de Salamanca, 2005.
- [15]. Moodle. Moodle. IMS Enterprise. . [En línea] 17 de mayo de 2006. [Citado el: 18 de febrero de 2009.] http://docs.moodle.org/es/IMS_Enterprise..
- [16]. JORUM & Project. The JISC Online Repository for [learning and teaching] Materials. . [En línea] 2004. [Citado el: 5 de marzo de 2009.] http://www.jorum.ac.uk/docs/Vol1_Fin.pdf..
- [17]. Downes, S. The Learning Marketplace. Meaning, Metadata and Content Syndication in the Learning Object Economy. Moncton, New Brunswick. [En línea] 2004. [Citado el: 15 de diciembre de 2008.] <http://www.downes.ca/files/book3.htm..>
- [18]. Rehak, D. & Mason, R. *Keeping the Learning in Learning Objects.* En A. Littlejohn (Ed), *Reusing Online Resources: A Sustainable Approach to E-learning (pp 20-34)*. London: Sterling : s.n., 2003.
- [19]. Borgman, C. L. *What are Digital Libraries, Who is Building them, and Why?* En Aparac, T. (Ed.), *Digital libraries: Interdisciplinary concepts, challenges and opportunities.* . Zagreb: Benja. : s.n., 1999.
- [20]. López, C. Modelo para el Desarrollo de Bibliotecas Digitales Especializadas. Tesis de Maestría, Instituto Tecnológico Autónomo de México. . [En línea] 2000. [Citado el: 2 de diciembre de 2008.] http://www.bibliodgsca.unam.mx/tesis/tes7cllg/sec_10.html.
- [21]. Peña, R. Baeza-Yates, R. & Rodríguez, J. V. *Gestión Digital de la Información, de Bits a Bibliotecas Digitales y la Web.* Madrid : RA-MA., 2002.
- [22]. E. SANCRISTOBAL, E. LÓPEZ-ALDEA , y otros. *DESARROLLO E INTEGRACIÓN DE LABORATORIOS VIRTUALES REMOTOS CON LOS SISTEMAS DE ADMINISTRACIÓN DE APRENDIZAJES ABIERTOS.* España : s.n., 2008.
- [23]. Rodríguez, Manuel Ángel. Perspectivas Microsoft. La nueva gestión académica. [En línea] 2003. [Citado el: 2 de febrero de 2009.] http://www.microsoft.com/spain/enterprise/perspectivas/numero_7/educacion.mspx..
- [24]. UNNE. *Sistema de Gestión Académica.* . 2005.
- [25]. Cabrera, Natavidad. Universida Oberta de Cataluña. La aplicación de las tecnologías en la gestión de comunidades virtuales de aprendizaje: la Secretaría virtual. [En línea] 2008. [Citado el: 15 de enero de 2009.] <http://www.ucm.es/info/multidoc/multidoc/revista/num8/cabrera.html#secretaria..>

- [26]. Peire, Juan, y otros. *Campus Virtual de la UNED: Un ejemplo de aplicación de nuevas tecnologías en la Educación a Distancia*. España : s.n., 1999.
- [27]. Bou Bauzá, Guillem. *Informe Comparativo*. Barcelona : UNED-UOC, 2001.
- [28]. Daudinot Founier, Isabel. *El Portal de la Red Cubana de la Ciencia. Codificando Dublin Core en XHTML*. . Ciudad de la habana : s.n., 2005.
- [29]. Secilia, Miguel Angel. Libro de Buenas Prácticas de e-learning. ESTÁNDARES DE E-LEARNING. . [En línea] 2007. [Citado el: 6 de febrero de 2009.] <http://www.buenaspracticaselearning.com/Colaboradores.html>..
- [30]. Furrie, Betty. Conociendo MARC Bibliográfico. Que es un Registro MARC . [En línea] 2003. [Citado el: 29 de febrero de 2009.] <http://www.loc.gov/marc/umbspa/um01a06.html>..
- [31]. Mcke, (Riley &. IMS Digital Repositories Interoperability-Core Functions Information Model . [En línea] 2003b. [Citado el: 3 de febrero de 2009.] http://www.imsglobal.org/digitalrepositories/driv1p0/imsdri_infov1p0.html. .
- [32]. IMS Enterprise Service. . *IMS Enterprise Service*. . s.l. : IMS Global Learning Consortium, 2004.
- [33]. IMS Global Learning Consortium. IMS Learner Information Packaging Information Model Specification [En línea] 26 de enero de 2009. [Citado el: 26 de febrero de 2009.] <http://www.imsglobal.org/profiles/index.html>..
34. Lagoze, Carl, y otros. *The Open Archives Initiative Protocol for Metadata Harvesting*. 2001.
- [35]. Agrega. Agrega. Interfaz SQI (Simple Query Interface). [En línea] 22 de septiembre de 2008. [Citado el: 25 de febrero de 2009.] <http://www.proyectoagrega.es/blog/?p=121>..
- [36]. Simon, Bernd, y otros. *Simple Query Interface, Specification*. 2005.
- [37]. OKI. Open Knowledge Initiative. OSID V2. . [En línea] 1 de noviembre de 2008. [Citado el: 20 de enero de 2009.] <http://www.okiproject.org/view/html/site/oki/node/8706>..
- [38]. Isaza E., Gustavo A. *Estándares de seguridad basados en XML*. 2007.
- [39]. W3C. World Wide Web Consortium . [En línea] W3C, 7 de Febrero de 2008. [Citado el: 26 de Febrero de 2009.]. *Guía Breve de Seguridad*. [En línea] febrero de 2008. [Citado el: 26 de febrerop de 2009.] <http://www.w3c.es/divulgacion/guiasbreves/Seguridad>..
- [40]. Menéndez Cisterna, David. Artículos en Scour Design. XML Encryption. . [En línea] 10 de diciembre de 2002. [Citado el: 26 de febrero de 2009.] http://www.scourdesign.com/articulos/xml_encryption.php..

- [41]. Network World. Network Word. [En línea] 1 de Abril de 2004. [Citado el: 25 de Febrero de 2009.] . *XrML: derechos digitales bajo control*. [En línea] 1 de abril de 2004. [Citado el: 25 de febrero de 2009.] <http://www.idg.es/comunicaciones/articulo.asp?id=157717..>
- [42]. Seguridad en Servicio Web. SAML. [En línea] 2006. [Citado el: 15 de febrero de 2009.] [http://www.di.uniovi.es/~labra/cursos/XML/XMLSecurity.html#\(18\)..](http://www.di.uniovi.es/~labra/cursos/XML/XMLSecurity.html#(18)..)
- [43]. McCabe, F, Haas,H, Booth, D. *Arquitectura Orientada a Servicio y Web Services: Conceptos, Tecnologías y Herramientas*. 2007.
- [44]. Lázaro Blanco, César. *Desarrollo de Aplicaciones SOA con WS-Security*. . 2005.
- [45]. Alvez, P, Foti, P y Scalone, M. *Estado Del Arte*. 2007.
- [46]. TIBCO. *El papel de un bus de servicios empresariales (ESB) en una SOA*. 2007.
- [47]. GONGO . *LABS, Preguntas que se hace un empresario ante la implantación de un BPM*. 2008 .
- [48]. Barriel Vistorial, Orlando. *Seguridad en Akademos 2.0*. . Ciudad habana : s.n., 2008.
- [49]. Ruiz Ortiz, Lidia, y otros. Entorno Virtual de Aprendizaje. *Selección de un Entorno Virtual de Aprendizaje para la Universidad de las Ciencias Informáticas*. [En línea] mayo de 2005. [Citado el: 5 de marzo de 2009.] <http://teleformacion.uci.cu/mod/resource/view.php?id=6..>
- [50]. Díaz Tito, Benjamin, Hidalgo Guillén, Jesús y Gutierrez, Yordanis. *Herramientas para la Reutilización de Contenidos a partir de la plataforma Moodle*. Ciudad Habana : s.n., 2008.
- [51]. OKI. *OSID V2 Specifications authentication package*. . Massachusetts : Massachusetts Institute of Technology, 2004.
- [52]. —. *OSID V2 Specifications authorization package*. Massachusetts : Massachusetts Institute of Technology, 2004.
- [53]. —. *OSID V3 Specifications loggin package*. EEUU : Massachusetts Institute of Technology, 2008.
- [54]. Carles Collazos, Josep y Cristofol, Gerard. *Convergencia a estándares eLearning para la UOC*. España : s.n., 2002.
- [55]. ANSI/NISO. *Information Retrieval (Z39.50):Application Service Definition and Protocol Specification*. Maryland : U.S.A. : National Information Standards Organization, 2003. ISSN: 1041-5653.
- [56]. Kaler, Chris y Lawrence, Kelvin. *Web Services Security*. . s.l. : OASIS, 2006.

BIBLIOGRAFÍA CONSULTADA

Alvez, P, Foti, P y Scalone, M. *Estado Del Arte*. 2007.

Agrega. Agrega. Interfaz SQL (Simple Query Interface). [En línea] 22 de septiembre de 2008. [Citado el: 25 de febrero de 2009.] <http://www.proyectoagrega.es/blog/?p=121..>

ANSI/NISO. *Information Retrieval (Z39.50):Application Service Definition and Protocol Specification*. .

AulaGlobal. 2005. Aula Global Observatorio de e-learning. Plataforma LMS. Plataforma de Formación. [En Línea] 28 de 02 de 2005. [Citado: 10 de 11 de 2008].

<http://www.aulaglobal.net.ve/observatorio/articles.php?lng=es&pg=156>

Barriel Vistorial, Orlando. *Seguridad en Akademos 2.0*. . Ciudad habana : s.n., 2008.

Biblioteca Digital. Servicios. [En línea] 2005. [Citado el: 15 de noviembre de 2008.] <http://biblioteca.uci.cu/bives/informacion.htm>.

BIDIPED. Biblioteca Digital para personas con Discapacidad. Glosario de Términos. Biblioteca Virtual. [En línea] 2007. [Citado el: 11 de noviembre de 2008.] <http://www.oit.or.cr/bidiped/Glosario.html>.

Borgman, C. L. *What are Digital Libraries, Who is Building them, and Why? En Aparac, T. (Ed.), Digital libraries: Interdisciplinary concepts, challenges and opportunities*. . Zagreb: Benja. : s.n., 1999.

Bou Bauzá, Guillem. *Informe Comparativo*. Barcelona : UNED-UOC, 2001.

Cabrera, Natavidad. Universida Oberta de Cataluña. La aplicación de las tecnologías en la gestión de comunidades virtuales de aprendizaje: la Secretaría virtual. [En línea] 2008. [Citado el: 15 de enero de 2009.] <http://www.ucm.es/info/multidoc/multidoc/revista/num8/cabrera.html#secretaria..>

Cañizares Glez, Roxana. *Framework para la gestión de contenidos educativos*. Ciudad de la Habana : s.n., 2008.

Carrodegas Argote, Judith y Ricardo Labañino, Henry.2008. *Análisis y Diseño de una herramienta de autor Web interoperable*. Ciudad de la Habana: s.n., 2008.

Cebrián, Manuel; Góngora Rojas, Andrea, Enseñanza virtual para la innovación universitaria, Madrid: Narcea, 2003.

CSIC. Ranking Web de Universidades del Mundo. Ranking Mundial. [En línea] 2008. [Citado el: 15 de diciembre de 2008.] http://www.webometrics.info/top4000_es.asp?offset=0..

Cuesta Mena, Dayteé. 2005. *NUEVAS APLICACIONES INTERACTIVAS PARA LA EDUCACIÓN A DISTANCIA*. [En línea] 2005. [Citado el: 11 de Febrero de 2009.]

<http://tecnologiaedu.us.es/ticsxxi/comunic/dcm.htm>.

Daudinot Founier, Isabel. *El Portal de la Red Cubana de la Ciencia. Codificando Dublin Core en XHTML*. . Ciudad de la habana : s.n., 2005.

Díaz Tito, Benjamin, Hidalgo Guillén, Jesús y Gutierrez, Yordanis. *Herramientas para la Reutilización de Contenidos a partir de la plataforma Moodle*. Ciudad Habana : s.n., 2008.

Downes, S. The Learning Marketplace. Meaning, Metadata and Content Syndication in the Learning Object Economy. Moncton, New Brunswick. [En línea] 2004. [Citado el: 15 de diciembre de 2008.] <http://www.downes.ca/files/book3.htm>..

El Pais. 2008. *El Pais.com*. Moodle llena la geografía educativa española de campus virtuales. [En línea] 4 de Abril de 2008. [Citado el: 11 de Febrero de 2009.]

http://www.elpais.com/articulo/portada/Moodle/llena/geografia/educativa/espanola/campus/virtuales/elpepi-supcib/20081204elpcibpor_1/Tes/

E. SANCRISTOBAL, E. LÓPEZ-ALDEA , y otros. *DESARROLLO E INTEGRACIÓN DE LABORATORIOS VIRTUALES REMOTOS CON LOS SISTEMAS DE ADMNISTRACIÓN DE APRENDIZAJES ABIERTOS*. España : s.n., 2008.

Estrada, V. H. M. and J. M. Nájeras.2005 “*Los Laboratorios Virtuales para la enseñanza formativa de los contenidos de ciencias naturales. Una experiencia en la Educación Superior a distancia de la UNED de Costa Rica 2005*”. [Disponible en:

<http://somi.cinstrum.unam.mx/virtualeduca2005/resumenes/2005-03-0136educa05labvirt.doc>

Furrie, Betty. Conociendo MARC Bibliográfico. Que es un Registro MARC . [En línea] 2003. [Citado el: 29 de febrero de 2009.] <http://www.loc.gov/marc/umbspa/um01a06.html>..

GONGO . *LABS, Preguntas que se hace un empresario ante la implantación de un BPM*. 2008 .

Hernández Ferrer, Kenia y Vázquez Mora, Miriolka. *Laboratorio Virtual de Física: Propuestas de nuevas Funcionalidades en la UCI*. Ciudad de la Habana, Cuba : s.n., 2007.

IMS Global Learning Consortium. IMS Learner Information Packaging Information Model Specification . . [En línea] 26 de enero de 2009. [Citado el: 26 de febrero de 2009.] <http://www.imsglobal.org/profiles/index.html>..

IMS Enterprise Service. . *IMS Enterprise Service*. . s.l. : IMS Global Learning Consortium, 2004.

Isaza E., Gustavo A. *Estándares de seguridad basados en XML*. 2007.

- Jardines Méndez, J. B. *Educación en red: mucho más que Educación a Distancia. Experiencias de las universidades médicas cubanas*. 2007.
- JORUM & Project. The JISC Online Repository for [learning and teaching] Materials. . [En línea] 2004. [Citado el: 5 de marzo de 2009.] http://www.jorum.ac.uk/docs/Vol1_Fin.pdf..
- Kaler, Chris y Lawrence, Kelvin. *Web Services Security*. . s.l. : OASIS, 2006.
- La Menza, Mario Alberto. Introducción a la arquitectura orientada a servicio. Introducción . [En línea] 2007. [Citado el: 8 de diciembre de 2008.] http://www.adeideas.com/paquito/archives/174/Arquitectura_orientada_servicio.htm.
- Lagoze, Carl, y otros. *The Open Archives Initiative Protocol for Metadata Harvesting*. 2001.
- Lázaro Blanco, César. *Desarrollo de Aplicaciones SOA con WS-Security*. . 2005.
- López, C. Modelo para el Desarrollo de Bibliotecas Digitales Especializadas. Tesis de Maestría, Instituto Tecnológico Autónomo de México. . [En línea] 2000. [Citado el: 2 de diciembre de 2008.] http://www.bibliodgsca.unam.mx/tesis/tes7cllg/sec_10.html.
- Lopez Guzmán, Clara. *Los Repositorios de Objetos de Aprendizaje como soporte a un entorno e-learning. Tesina doctoral*. España, Salamanca : Universidad de Salamanca, 2005.
- López Neira., Alexander. 2007. IPLAC VIRTUAL. [En Línea] 2007. [Citado el: 3 de 11 de 2008.] <http://www.iplacvirtual.rimed.cu/iplacvirtual.htm>
- Martí Arias, José. INTRODUCCIÓN Y DESARROLLO DEL E-LEARNING EN EL CENTRO DE ESTUDIO DE ADMINISTRACIÓN PÚBLICA. [En línea] 2004. [Citado el: 3 de marzo de 2009.] http://74.125.47.132/search?q=cache:B_mmqNmYQN4J:www.clad.org.ve/documentos/otros-documentos/marti+desarrollo+de+la+inform%C3%A1tica+%2B+desarrollando+el+e-learning&hl=es&ct=clnk&cd=8&gl=cu..
- Maryland : U.S.A. : National Information Standards Organization, 2003. ISSN: 1041-5653..
- Menéndez Cisterna, David. Artículos en Scour Design. XML Encryption. . [En línea] 10 de diciembre de 2002. [Citado el: 26 de febrero de 2009.] http://www.scourdesign.com/articulos/xml_encryption.php..
- Moodle. Moodle. IMS Enterprise. . [En línea] 17 de mayo de 2006. [Citado el: 18 de febrero de 2009.] http://docs.moodle.org/es/IMS_Enterprise..
- Mcke & Riley. IMS Digital Repositories Interoperability-Core Functions Information Model . [En línea] 2003b. [Citado el: 3 de febrero de 2009.] http://www.imsglobal.org/digitalrepositories/driv1p0/imsdri_infov1p0.html. .

- McCabe, F, Haas,H, Booth, D. *Arquitectura Orientada a Servicio y Web Services: Conceptos, Tecnologías y Herramientas*. 2007.
- Microsoft.TechNet. 2003. Microsoft.TechNet. *Descripción de los certificados digitales*. [En línea] 2003. [Citado el: 16 de febrero de 2009.] [http://technet.microsoft.com/es-es/library/bb123848\(EXCHG.65\).aspx](http://technet.microsoft.com/es-es/library/bb123848(EXCHG.65).aspx).
- Network World. Network Word. [En línea] 1 de Abril de 2004. [Citado el: 25 de Febrero de 2009.] . *XrML: derechos digitales bajo control*. [En línea] 1 de abril de 2004. [Citado el: 25 de febrero de 2009.] <http://www.idg.es/comunicaciones/articulo.asp?id=157717..>
- OKI. Open Knowledge Initiative. OSID V2. . [En línea] 1 de noviembre de 2008. [Citado el: 20 de enero de 2009.] <http://www.okiproject.org/view/html/site/oki/node/8706..>
- OKI. *OSID V2 Specifications authentication package*. . Massachusetts : Massachusetts Institute of Technology, 2004.
- OKI. *OSID V2 Specifications authorization package*. Massachusetts : Massachusetts Institute of Technology, 2004.
- OKI. *OSID V3 Specifications loggin package*. EEUU : Massachusetts Institute of Technology, 2008.
- Carles Collazos, Josep y Cristofol, Gerard. *Convergencia a estándares eLearning para la UOC*. España : s.n., 2002.
- Peire, Juan, y otros. *Campus Virtual de la UNED: Un ejemplo de aplicación de nuevas tecnologías en la Educación a Distancia*. España : s.n., 1999.
- Peña, R. Baeza-Yates, R. & Rodríguez, J. V. *Gestión Digital de la Información, de Bits a Bibliotecas Digitales y la Web*. Madrid : RA-MA., 2002.
- Pérez, Adoració, Serrano, Jordi y Enrech, Marta. 1998. FESABID 98 VI Jornadas Españolas de Documentación . *UNA BIBLIOTECA VIRTUAL PARA UNA COMUNIDAD VIRTUAL*. [En línea] 1998. [Citado el: 10 de Marzo de 2009.] http://fesabid98.florida-uni.es/Comunicaciones/a_perez/A_Perez.htm.
- Red TTnet. *La formación sin distancia. Estudio realizado por el Grupo de Trabajo de e-Learning de la red TTnet* . 2005.
- Rehak, D. & Mason, R. *Keeping the Learning in Learning Objects*.En A. Littlejohn (Ed), *Reusing Online Resources: A Sustainable Approach to E-learning (pp 20-34)*. London: Sterling : s.n., 2003.
- Rodríguez, Manuel Ángel. *Perspectivas Microsoft. La nueva gestión académica*. [En línea] 2003. [Citado el: 2 de febrero de 2009.] http://www.microsoft.com/spain/enterprise/perspectivas/numero_7/educacion.mspx..

- Ruiz Ortiz, Lidia, y otros. Entorno Virtual de Aprendizaje. *Selección de un Entorno Virtual de Aprendizaje para la Universidad de las Ciencias Informáticas*. [En línea] mayo de 2005. [Citado el: 5 de marzo de 2009.] <http://teleformacion.uci.cu/mod/resource/view.php?id=6..>
- Salas Campos, Ileana. 2008. *Procesos de virtualización en la UNED de Costa Rica*. [En línea] 2008. [Citado el: 11 de Febrero de 2009.] http://209.85.173.132/search?q=cache:YuV9_noSACMJ:www.virtualeduca.info/ponencias/211/Procesos%2520de%2520virtualizaci%25F3n%2520en%2520la%2520UNED%2520de%2520Costa%2520Rica.doc+uned+de+costa+rica&hl=es&ct=clnk&cd=7&gl=cu
- Secilia, Miguel Angel. Libro de Buenas Prácticas de e-learning. ESTÁNDARES DE E-LEARNING. . [En línea] 2007. [Citado el: 6 de febrero de 2009.] <http://www.buenaspracticaselearning.com/Colaboradores.html..>
- Seguridad en Servicio Web. SAML. [En línea] 2006. [Citado el: 15 de febrero de 2009.] [http://www.di.uniovi.es/~labra/cursos/XML/XMLSecurity.html#\(18\)..](http://www.di.uniovi.es/~labra/cursos/XML/XMLSecurity.html#(18)..)
- Simon, Bernd, y otros. *Simple Query Interface, Specification*. 2005.
- Tamayo Ávila, D. *Herramientas para la reutilización de contenidos educativos* . 2007.
- TIBCO. *El papel de un bus de servicios empresariales (ESB) en una SOA*. 2007.
- Universidad Virtual UTM. Universidad Virtual de la Maxteca. . [En línea] 2006. [Citado el: 10 de noviembre de 2008.] <http://virtual.utm.mx/>.
- UNNE. *Sistema de Gestión Académica*. . 2005.
- USMPVirtual. Universidad Virtual. . [En línea] 2006. [Citado el: 30 de enero de 2009.] [http://www.usmpvirtual.edu.pe/..](http://www.usmpvirtual.edu.pe/)
- UIB.2008. Universidad de Islas Baleares. *Universidad de Islas Baleares*. [En Línea] 2008. [Citado el: 07 de 12 de 2008.] <http://www.uib.es/ca/>
- UVS Cubana. 2001. UVS Cubana. Universidad Virtual de Salud Cubana. [En Línes] 2001. [Citado el: 9 de 11 de 2008] <http://www.uvirtual.sld.cu/>
- W3C. World Wide Web Consortium . [En línea] W3C, 7 de Febrero de 2008. [Citado el: 26 de Febrero de 2009.]. *Guía Breve de Seguridad*. [En línea] febrero de 2008. [Citado el: 26 de febrerop de 2009.] <http://www.w3c.es/divulgacion/guiasbreves/Seguridad..>

Anexo 1. Entrevista

Fecha: 24/01/2009 Hora: 2:30 pm

Jefa del Proyecto Akademos

¿Cuáles son las principales funcionalidades que brinda en la actualidad?

¿Sobre qué estándares se rige o trabaja la última versión disponible en la UCI? ¿Utiliza algún estándar de interoperabilidad?

¿Hasta qué punto es interoperable con algunos de los componentes como el EVA, La Biblioteca Virtual o Digital, etc. de una UV?

¿Está capacitado para integrarse con otras herramientas? ¿Cuáles?

¿Aprovecha las ventajas que ofrece una arquitectura orientada a servicio?

Anexo 2. Entrevista

Fecha: 10/02/2009 Hora: 2:30 pm

Jefa del Proyecto de Biblioteca Automatizada de la UCI

¿Cuáles son las principales funcionalidades que brinda en la actualidad?

¿Sobre qué estándares se rige o trabaja la última versión disponible en la UCI? ¿Utiliza algún estándar de interoperabilidad?

¿Hasta qué punto es interoperable con algunos de los componentes como el EVA, La Biblioteca Virtual o Digital, etc. de una UV?

¿Está capacitado para integrarse con otras herramientas? ¿Cuáles?

¿Aprovecha las ventajas que ofrece una arquitectura orientada a servicio?

Anexo 3. Otros métodos del estándar OSID Authentication

Iteradores

Iteradores devuelve un conjunto de elementos de un tipo específico, de uno en uno. El objetivo de todos los iteradores es ofrecer un camino para los métodos OSID para retornar los valores múltiples de un tipo común y no utilizar una matriz. Retornando un conjunto pueden no ser apropiados si el número de valores que se muestra es grande o es remotamente descabellado. Iteradores no permitir el acceso a los valores de índice, y no debe acceder a los valores de secuencia. Asimismo, no hay forma de dar marcha atrás a través de la secuencia, si no que los valores queden en una estructura de datos, como un matriz, que permite el acceso por índice.

Tipo de Retorno	Método	Descripción
boolean	hasNext<Object type>	Método que devuelve true si hay más valores disponibles del tipo iterador; sino false.
	next<Object type> ()	Método que devuelve el elemento siguiente en la secuencia. Tenga en cuenta que en muchos casos, el orden de los elementos no está garantizado.

Tabla 57. Otros métodos del estándar OSID Authentication

Anexo 4. Otros métodos del estándar OSID Authorization

osid.authorization.Authorization

La interfaz autorización se conecta al AgentId, a Function, y a Qualifier. Estos son todos los pasó dentro del método createFunction () en el Manager. Las autorizaciones tienen una fecha de entrada en vigor y una de vencimiento. Estos opcionalmente es pasado dentro de createFunction (). La fecha de entrada en vigor y de vencimiento puede ser actualizado. La implementación de la OSID autorización determina las actualizaciones modifiedBy, isActiveNow y isExplicit.

Tipo de Retorno	Método
long	getEffectiveDate()
long	getExpirationDate()
Id	getModifiedBy()
long	getModifiedDate()
Function	getFunction()
Qualifier	getQualifier()
Id	getAgentId()
boolean	isActiveNow()
boolean	isExplicit()
void	updateExpirationDate()
void	updateEffectiveDate()

Tabla 58. *Métodos de la interfaz osid.authorization.Authorization*

org.osid.authorization.Function

La interfaz function simplemente se refiere a los atributos de Function. Las funciones tienen un nombre de referencia inmutable, una descripción, un FunctionType, un QualifiedHierarchyId, y un ID. Estos son todos pasados dentro del método createFunction () en el Manager. La descripción puede ser actualizada.

Tipo de Retorno	Método
String	getDescription()
String	getReferenceName()
Type	getFunctionType()
Id	getId()
Id	getQualifierHierarchyId()
Void	updateDescription(String description)

Tabla 59. Métodos de la interfaz *org.osid.authorization.Function*

org.osid.authorization.Qualifier

La interfaz Qualifier (Calificador) se refiere a los atributos de la clasificación, así como a los métodos para gestionar e interrogar la jerarquía de clasificación. Los Calificadores tienen un nombre, una descripción, un QualifiedType, y un ID. Estos son todos pasados dentro de los métodos createQualifier () o createRootQualifier () en el Administrador. El Indicar nombre y la descripción se puede cambiar, mientras que los otros atributos no.

La jerarquía de Qualifier es compuesta y mantenida mediante los métodos addParent (), changeParent (), y removeParent (). La jerarquía de Qualifier es interrogado utilizando isParent (), getChildren (), isChildOf () e isDescendantOf (). Para obtener todos los descendientes, llame getChildren () recursivamente.

Tipo de Retorno	Método	Parámetros
void	addParent()	Id parentQualifierId
void	changeParent()	Id oldParentId, Id newParentId
QualifierIterator	getChildren()	
String	getDescription()	
String	getReferenceName()	
Id	getId()	
QualifierIterator	getParents()	
Type	getQualifierType()	

boolean	isChildOf()	Id parentId
boolean	isDescendentOf()	Id ancestorId
boolean	isParent()	
void	removeParent()	Id parentQualifierId
void	updateDescription()	String description

Tabla 60. Métodos de la interfaz *org.osid.authorization.Qualifier*

org.osid.authorization.Object Iterators

Iteradores devuelve un conjunto de elementos de un tipo específico, de uno en uno. El objetivo de todos los iteradores es ofrecer un camino para los métodos OSID para retornar los valores múltiples de un tipo común y no utilizar una matriz. Retornando un conjunto pueden no ser apropiados si el número de valores que se muestra es grande o es remotamente descabellado. Iteradores no permitir el acceso a los valores de índice, y no debe acceder a los valores de secuencia. Asimismo, no hay forma de dar marcha atrás a través de la secuencia, si no que los valores queden en una estructura de datos, como un matriz, que permite el acceso por índice.

Tipo de Retorno	Método	Descripción
boolean	hasNext<Object type>	Método que devuelve true si hay más valores disponibles del tipo iterador; sino false.
	next<Object type> ()	Método que devuelve el elemento siguiente en la secuencia. Tenga en cuenta que en muchos casos, el orden de los elementos no está garantizado.

Tabla 61. Métodos de la interfaz *org.osid.authorization.Object Iterators*

Anexo 5. Otros métodos del estándar OSID Logging

Interfaz osid.logging.LoggingManager

Iterators		
Método	getLoggingSession	
Descripción	Obtiene la OsidSession asociada con el servicio.	
Tipo de Retorno	osid.logging.LoggingSession una LoggingSession	
Errores	UNIMPLEMENTED supportsLogging() es falso OPERATION_FAILED incapaz de completar la solicitud	
Iterators		
Método	getLoggingSessionForLog	
Descripción	Obtiene la OsidSession asociada con el servicio logging para el log dado.	
Tipo de Retorno	osid.logging.LoggingSession una LoggingSession	
Parámetros	osid.id.Id	logId El Id del Log
Errores	NOT_FOUND no hay Log encontrado para el Id dado. NULL_ARGUMENT logId es nulo OPERATION_FAILED incapaz de completar la solicitud UNIMPLEMENTED supportsLogging() o supportsVisibleFederation() es falso	
Iterators		
Método	getLogReadingSession	
Descripción	Obtiene la OsidSession asociada con el servicio logging de lectura	
Tipo de Retorno	osid.logging.LogReadingSession una LogReadingSession	
Errores	OPERATION_FAILED incapaz de completar la solicitud UNIMPLEMENTED supportsLogReading() es falso	
Iterators		
Método	getLogReadingSessionForLog	
Descripción	Obtiene la OsidSession asociada con el servicio logging de lectura para un log	

	dato.		
Parámetros	osid.id.Id	logId	el Id del Log
Tipo de Retorno	osid.logging.LogReadingSession una LogReadingSession		
Errores	NOT_FOUND no hay Log encontrado para el Id dado NULL_ARGUMENT logId es nulo OPERATION_FAILED incapaz de completar la solicitud UNIMPLEMENTED supportsLogReading() o supportsVisibleFederation() es falso		
Método			
	getLogEntrySearchSession		
Descripción			
	Obtiene la OsidSession asociada con el servicio logging de entrada de búsqueda.		
Tipo de Retorno			
	osid.logging.LogEntrySearchSession una LogEntrySearchSession		
Errores			
	OPERATION_FAILED incapaz de completar la solicitud UNIMPLEMENTED supportsLogEntrySearch() es falso.		
Método			
	getLogEntrySearchSessionForLog		
Descripción			
	Obtiene la OsidSession asociada con el servicio logging de entrada de búsqueda para el log dado.		
Parámetros	osid.id.Id	logId	el Id del Log
Tipo de Retorno			
	osid.logging.LogEntrySearchSession una LogEntrySearchSession		
Errores			
	NOT_FOUND no hay Log encontrado para el Id dado NULL_ARGUMENT logId es nulo OPERATION_FAILED incapaz de completar la solicitud UNIMPLEMENTED supportsLogEntrySearch() o supportsVisibleFederation() es falso.		
Método			
	getLogEntryAdminSession		
Descripción			
	Obtiene la OsidSession asociada con el servicio logging de entrada administrativo.		
Tipo de Retorno			
	osid.logging.LogEntryAdminSession una LogEntryAdminSession		
Errores			
	OPERATION_FAILED incapaz de completar la solicitud		

	UNIMPLEMENTED supportsLogEntryAdmin() es falso.		
Método	getLogEntryAdminSessionForLog		
Descripción	Obtiene la OsidSession asociada con el servicio logging de entrada administrativo para un log dado.		
Tipo de Retorno	osid.logging.LogEntryAdminSession una LogEntryAdminSession		
Errores	NOT_FOUND no hay Log encontrado para el Id dado NULL_ARGUMENT logId es nulo. OPERATION_FAILED incapaz de completar la solicitud UNIMPLEMENTED supportsLogEntryAdmin() o supportsVisibleFederation() es falso		
Método	getLogEntryNotificationSession		
Descripción	Obtiene la OsidSession asociada con el servicio logging de entrada de notificación.		
Tipo de Retorno	osid.logging.LogEntryNotificationSession una LogEntryNotificationSession		
Errores	OPERATION_FAILED incapaz de completar la solicitud UNIMPLEMENTED supportsLogEntryNotification() es falso.		
Método	getLogEntryNotificationSessionForLog		
Descripción	Obtiene la OsidSession asociada con el servicio logging de entrada de notificación para un log dado.		
Parámetros	osid.id.Id	logId	el Id del Log
Tipo de Retorno	osid.logging.LogEntryNotificationSession una LogEntryNotificationSession		
Errores	NOT_FOUND no hay Log encontrado para el Id dado NULL_ARGUMENT logId es nulo. OPERATION_FAILED incapaz de completar la solicitud UNIMPLEMENTED supportsLogEntryNotification() o supportsVisibleFederation() es falso.		
Método	getLogLookupSession		

Descripción	Obtiene la OsidSession asociada con el servicio log de búsqueda.
Tipo de Retorno	osid.logging.LogLookupSession una LogLookupSession
Errores	OPERATION_FAILED incapaz de completar la solicitud UNIMPLEMENTED supportsLogLookup() es falso.
Método	
Descripción	Obtiene el OsidSession asociados con el servicio del registro administrativo
Tipo de Retorno	osid.logging.LogAdminSession una LogAdminSession
Errores	OPERATION_FAILED incapaz de completar la solicitud UNIMPLEMENTED supportsLogAdmin() es falso
Método	
Descripción	Obtiene la OsidSession asociada con el servicio del registro de notificación
Tipo de Retorno	osid.logging.LogNotificationSession una LogNotificationSession
Errores	OPERATION_FAILED incapaz de completar la solicitud UNIMPLEMENTED supportsLogNotification() es falso.
Método	
Descripción	Obtiene la OsidSession asociada con el servicio del registro de la jerarquía.
Tipo de Retorno	osid.logging.LogHierarchySession una LogHierarchySession por logs
Errores	OPERATION_FAILED incapaz de completar la solicitud UNIMPLEMENTED supportsLogHierarchy() es falso
Método	
Descripción	Obtiene la OsidSession asociada con el servicio del registro de la jerarquía de diseño.
Tipo de Retorno	osid.logging.LogHierarchyDesignSession una HierarchyDesignSession por registros
Errores	OPERATION_FAILED incapaz de completar la solicitud UNIMPLEMENTED supportsLogHierarchyDesign() es falso

Tabla 62. Métodos de la interfaz `osid.logging.LoggingManager`**Interfaz `osid.logging.LoggingProxyManager`**

Método	getLoggingSession		
Descripción	Obtiene la OsidSession asociada con el servicio logging		
Parámetros	osid.authentication.Authentication	authentication	Autetificación del proxy
Tipo de Retorno	osid.logging.LoggingSession una LoggingSession		
Errores	NULL_ARGUMENT la autenticación es nula OPERATION_FAILED incapaz de completar la solicitud PERMISSION_DENIED la autenticación no es válido UNIMPLEMENTED supportsLogging() es falso UNSUPPORTED el servicio de autenticación no es compatible		
Método	getLoggingSessionForLog		
Descripción	Obtiene la OsidSession asociada con el servicio logging para un registro dado.		
Parámetros	osid.authentication.Authentication	authentication	Autenticación del proxy
	osid.id.Id	logId	el Id del Log
Tipo de Retorno	osid.logging.LoggingSession una LoggingSession		
Errores	NOT_FOUND no se encontró el registro dado por el Id. NULL_ARGUMENT logId o la autenticación es nulo. OPERATION_FAILED incapaz de completar la solicitud PERMISSION_DENIED la autenticación no es válida UNIMPLEMENTED supportsLogging() o supportsVisibleFederation() es falso UNSUPPORTED el servicio de autenticación no es compatible.		
Método	getLogReadingSession		
Descripción	Obtiene la OsidSession asociada con el servicio de registro de la lectura		

Parámetros	osid.authentication.Authentication	authentication	Autenticación del proxy
Tipo de Retorno	osid.logging.LogReadingSession una LogReadingSession		
Errores	NULL_ARGUMENT autenticación nula OPERATION_FAILED incapaz de completar la solicitud PERMISSION_DENIED la autenticación no es válida UNIMPLEMENTED supportsLogReading() es falso UNSUPPORTED el servicio de autenticación no es compatible.		
Método			
Método	getLogReadingSessionForLog		
Descripción	Obtiene la OsidSession asociada con el servicio de registro de la lectura para un registro dado.		
Parámetros	osid.authentication.Authentication	authentication	Autenticación del proxy
	osid.id.Id	logId	el Id del Log
Tipo de Retorno	osid.logging.LogReadingSession una LogReadingSession		
Errores	NOT_FOUND no se encontró el registro dado por el Id. NULL_ARGUMENT logId la autenticación es nulo OPERATION_FAILED incapaz de completar la solicitud PERMISSION_DENIED la autenticación no es válida UNIMPLEMENTED supportsLogReading() o supportsVisibleFederation() es falso UNSUPPORTED el servicio de autenticación no es compatible.		
Método			
Método	getLogEntrySearchSessionForLog		
Descripción	Obtiene la OsidSession asociada con el servicio del registro de entrada de búsqueda para un registro dado.		
Parámetros	osid.authentication.Authentication	authentication	Autenticación del proxy
	osid.id.Id	logId	el Id del Log
Tipo de Retorno	osid.logging.LogEntrySearchSession una LogEntrySearchSession		
Errores	NOT_FOUND no se encontró el registro dado por el Id NULL_ARGUMENT logId o autenticación nula		

	<p>OPERATION_FAILED incapaz de completar la solicitud</p> <p>PERMISSION_DENIED autenticación no es válida.</p> <p>UNIMPLEMENTED supportsLogEntrySearch() o supportsVisibleFederation() es falso</p> <p>UNSUPPORTED el servicio de autenticación no es compatible.</p>		
Método	getLogEntryAdminSession		
Descripción	Obtiene la OsidSession asociada con el servicio del registro de entrada administrativo.		
Parámetros	osid.authentication.Authentication	authentication	Autenticación del proxy
Tipo de Retorno	osid.logging.LogEntryAdminSession una LogEntryAdminSession		
Errores	<p>NULL_ARGUMENT autenticación nula</p> <p>OPERATION_FAILED incapaz de completar la solicitud</p> <p>PERMISSION_DENIED logId o autenticación nula</p> <p>UNIMPLEMENTED supportsLogEntryAdmin() es falso.</p> <p>UNSUPPORTED el servicio de autenticación no es compatible</p>		
Método	getLogEntryAdminSessionForLog		
Descripción	Obtiene la OsidSession asociada con el servicio del registro de entrada administrativo para un log dado.		
Parámetros	osid.authentication.Authentication	authentication	Autenticación del proxy
	osid.id.Id	logId	el Id del Log
Tipo de Retorno	osid.logging.LogEntryAdminSession una LogEntryAdminSession		
Errores	<p>NOT_FOUND no se encontró el registro dado por el Id</p> <p>NULL_ARGUMENT logId o autenticación nula</p> <p>OPERATION_FAILED incapaz de completar la solicitud</p> <p>PERMISSION_DENIED autenticación no es válida.</p> <p>UNIMPLEMENTED supportsLogEntryAdmin() o supportsVisibleFederation() es falso</p> <p>UNSUPPORTED el servicio de autenticación no es compatible</p>		

Método	getLogEntryNotificationSession		
Descripción	Obtiene la OsidSession asociada con el servicio logging de entrada de notificación.		
Parámetros	osid.authentication.Authentication	authentication	Auenticación del proxy
Tipo de Retorno	osid.logging.LogEntryNotificationSession una LogEntryNotificationSession		
Errores	NULL_ARGUMENT autenticación nula OPERATION_FAILED incapaz de completar la solicitud PERMISSION_DENIED autenticación no es válida UNIMPLEMENTED supportsLogEntryNotification() es falso UNSUPPORTED el servicio de autenticación no es compatible		
Método	getLogEntryNotificationSessionForLog		
Descripción	Obtiene la OsidSession asociada con el servicio logging de entrada de notificación para un log dado.		
Parámetros	osid.authentication.Authentication	authentication	Autenticaión del proxy
	osid.id.Id	logId	el Id del Log
Tipo de Retorno	osid.logging.LogEntryNotificationSession una LogEntryNotificationSession		
Errores	NOT_FOUND no se encontró el registro dado por el Id NULL_ARGUMENT logId o autenticación nula OPERATION_FAILED incapaz de completar la solicitud PERMISSION_DENIED autenticación no es válida UNIMPLEMENTED supportsLogEntryNotification() o supportsVisibleFederation() es falso UNSUPPORTED el servicio de autenticación no es compatible		
Método	getLogLookupSession		
Descripción	Obtiene la OsidSession asociada con el servicio de registro de búsqueda		
Parámetros	osid.authentication.Authentication	authentication	autenticación del proxy
Tipo de Retorno	osid.logging.LogLookupSession una LogLookupSession		
Errores	NULL_ARGUMENT autenticación nula		

	<p>OPERATION_FAILED incapaz de completar la solicitud</p> <p>PERMISSION_DENIED autenticación no es válida</p> <p>UNIMPLEMENTED supportsLogLookup() es falso</p> <p>UNSUPPORTED el servicio de autenticación no es compatible</p>		
Método	getLogSearchSession		
Descripción	Obtiene el OsidSession asociados con el registro de servicio de búsqueda.		
Parámetros	osid.authentication.Authentication	authentication	autenticación del proxy
Tipo de Retorno	osid.logging.LogSearchSession una LogSearchSession		
Errores	<p>NULL_ARGUMENT autenticación nula</p> <p>OPERATION_FAILED incapaz de completar la solicitud</p> <p>PERMISSION_DENIED autenticación no es válida</p> <p>UNIMPLEMENTED supportsLogSearch() es falso</p> <p>UNSUPPORTED el servicio de autenticación no es compatible</p>		
Método	getLogNotificationSession		
Descripción	Obtiene el OsidSession asociados con el servicio de notificación de registro.		
Parámetros	osid.authentication.Authentication	authentication	autenticación del proxy
Tipo de Retorno	osid.logging.LogNotificationSession una LogNotificationSession		
Errores	<p>NULL_ARGUMENT autenticación nula</p> <p>OPERATION_FAILED incapaz de completar la solicitud</p> <p>PERMISSION_DENIED autenticación no es válida</p> <p>UNIMPLEMENTED supportsLogNotification() es falso</p> <p>UNSUPPORTED el servicio de autenticación no es compatible</p>		
Método	getLogHierarchySession		
Descripción	Obtiene el OsidSession asociados con el servicio de registro de la jerarquía.		
Parámetros	osid.authentication.Authentication	authentication	autenticación del proxy
Tipo de Retorno	osid.logging.LogHierarchySession una LogHierarchySession for logs		

Errores	NULL_ARGUMENT autenticación nula OPERATION_FAILED incapaz de completar la solicitud PERMISSION_DENIED autenticación no es válida UNIMPLEMENTED supportsLogHierarchy() es falso UNSUPPORTED el servicio de autenticación no es compatible		
Método	getLogHierarchyDesignSession		
Descripción	Obtiene la OsidSession asociada con el servicio de diseño de la jerarquía de log		
Parámetros	osid.authentication.Authentication	authentication	autenticación del proxy
Tipo de Retorno	osid.logging.LogHierarchyDesignSession una HierarchyDesignSession for logs		
Errores	NULL_ARGUMENT autenticación nula OPERATION_FAILED incapaz de completar la solicitud PERMISSION_DENIED autenticación no es válida UNIMPLEMENTED supportsLogHierarchyDesign() es falso UNSUPPORTED el servicio de autenticación no es compatible		

Tabla 63. Métodos de la Interfaz *osid.logging.LoggingProxyManager*

Interfaz *osid.logging.LogReadingSession*

Método	canReadLog
Descripción	Prueba si este usuario puede leer los log.El retorno de true no garantiza éxito en la autorización.Retornar falso indica que todos los métodos conocidos en la sesión resultarán en un PERMISSION_DENIED.
Tipo de Retorno	Boolean falso si los métodos leídos no están autorizados, sino true.
Método	useComparativeLogEntryView
Descripción	Lo retornado desde los métodos de búsqueda pueden omitir o traducir elementos basados en esta sesión tales como Descripción de la autorización y no resulta un error. Esta vista es utilizada cuando una gran interoperabilidad es deseada en la

	expensa de la precisión.	
Método	usePlenaryLogEntryView	
Descripción	Una vista complete de los LogEntry devueltos es deseada. Los métodos que devolverán lo solicitado o resultan en un error. La vista es utilizada cuando una gran precisión es deseada a expensas de la interoperabilidad.	
Método	useFederatedLogView	
Descripción	Federa las vistas para métodos en esta sesión. Una vista federada incluirá entrada de logs los cuales son hijos de este log en la jerarquía de log.	
Método	useIsolatedLogView	
Descripción	Aislar la vista para métodos en esta sesión. Una vista aislada restringe recuperaciones a este log solamente.	
Método	getLogEntry	
Descripción	Obtiene los LogEntry especificado por su id. En modo plenario, el id exacto es encontrado o un resultado NOT_FOUND. En otro caso, el LogEntry retornado puede tener un Id diferente al solicitado, tal como el caso donde un Id duplicado fue asignado a un LogEntry y mantenido para compatibilidad.	
Parámetros	osid.id.Id	logEntryId el id del LogEntry a recuperar
Tipo de Retorno	osid.logging.LogEntry retornado LogEntry	
Errores	NOT_FOUND logEntry no ha encontrado el Id dado NULL_ARGUMENT logEntryId es nula OPERATION_FAILED incapaz de completar la solicitud PERMISSION_DENIED autorización es un fracaso	
Método	getLogEntriesByIds	
Descripción	Obtiene una LogEntryList correspondiente al IdList dado. En modo plenario la lista	

	retornada contiene todas las entradas especificadas en las listas Id, en el orden de la lista incluyendo duplicados o un resultado de error, si un Id en la lista proveida no es encontrado o inaccesible. En otro caso, logEntries inaccesibles pueden ser omitidas de la lista y pueden presentar los elementos en cualquier orden devolviendo un conjunto único.		
Parámetros	osid.id.IdList	logEntryIdList	la lista de Id a recuperar
Tipo de Retorno	osid.logging.LogEntryList la lista LogEntry retornada		
Errores	NOT_FOUND un ID no fue encontrado NULL_ARGUMENT logEntryIdList es nulo OPERATION_FAILED incapaz de completar la solicitud PERMISSION_DENIED autorización es un fracaso		
Método			
	getLogEntriesByContentType		
Descripción	Obtiene una LogEntryList correspondiente al tipo de contenido dado. En modo plenario la lista retornada contiene todas las entradas conocidas o un resultado de error. En otro caso, la lista retornada puede contener solo aquellas entradas que son accesibles a través de esta sesión, en ambos casos el orden de los conjuntos no es especificado.		
Parámetros	osid.type.Type	contentType	Un log de engtrad de tipo de contenido
Tipo de Retorno	osid.logging.LogEntryList la lista LogEntry retornda		
Errores	NULL_ARGUMENT contentType es nulo OPERATION_FAILED incapaz de completar la solicitud PERMISSION_DENIED autorización es un fracaso		
Método			
	getLogEntriesByPriorityType		
Descripción	Obtiene una LogEntryList correspondiente al tipo de prioridad dada. En plenaria la lista devuelta contiene todas las entrads conocidas o un resultado de error. En otro caso la lista retornada puede contener solo aquellas entradas que son accesibles		

	mediante esta sesión. En ambos casos el orden de los conjuntos no es especificado		
Parámetros	osid.type.Type	priorityType	un tipo de prioridad de un log de entrada
Tipo de Retorno	osid.logging.LogEntryList la lista LogEntry retornada		
Errores	NULL_ARGUMENT contentType es nulo OPERATION_FAILED incapaz de completar la solicitud PERMISSION_DENIED autorización es un fracaso		
Método			
	getLogEntriesByTime		
Descripción	Obtiene una LogEntryList correspondiente al interval de tiempo dado. En plenaria la lista retornada contiene todas las entrads conocidas o un resultado de error. En otro caso la lista retornada puede contener solo aquellas entradas que son accesibles mediante esta sesión. En ambos casos el orden de los conjuntos no es especificado.		
Parámetros	osid.calendar.Date	Time	poner fin a una hora de inicio
	osid.calendar.DateTime	start	Un tiempo de arranque
Tipo de Retorno	osid.logging.LogEntryList la lista LogEntry retornada		
Errores	INVALID_ARGUMENT inicio es superior a fin OPERATION_FAILED incapaz de completar la solicitud PERMISSION_DENIED autorización es un fracaso		
Método			
	getLogEntries		
Descripción	Obtiene una LogEntryList correspondiente al interval de tiempo dado. En plenaria la lista retornada contiene todas las entrads conocidas o un resultado de error. En otro caso la lista retornada puede contener solo aquellas entradas que son accesibles mediante esta sesión. En ambos casos el orden de los conjuntos no es especificado.		
Tipo de Retorno	osid.logging.LogEntryList una lista de entradas de log		
Errores	OPERATION_FAILED incapaz de completar la solicitud PERMISSION_DENIED autorización es un fracaso		

Tabla 64. Métodos de la interfaz osid.logging.LogReadingSession

Interfaz osid.logging.LogEntrySearchSession

Método	canSearchLogEntries
Descripción	Prueba si este usuario puede realizar búsquedas LogEntry. Retornar verdadero no garantiza éxito en la autorización, falso indica que son conocidos todos los métodos en esta sesión que resultarán en un PERMISSION_DENIED. Esto es como una pista para una aplicación que pueden optar por no ofrecer a las operaciones de búsqueda a los usuarios no autorizados
Tipo de Retorno	Boolean, falso si los métodos de búsqueda no son autorizados, sino verdadero.
Método	useFederatedLogView
Descripción	Federa la vista para métodos en esta sesión. Una vista federada incluyen las entradas de log que son hijos de este log en el log de jerarquía.
Método	uselolatedLogView
Descripción	Aisla la vista para los métodos de esta sesión. Una opinión aislada restringe las búsquedas a solo este registro
Método	getLogEntryQuery
Descripción	Obtiene un log de entrada de la interfaz de búsqueda.
Tipo de Retorno	osid.logging.LogEntryQuery la entrada del log de búsqueda
Método	getLogEntriesByQuery
Descripción	Obtiene una lista de entradas del log que coinciden con el interfaz de búsqueda
Parámetros	osid.logging.LogEntryQuery logEntryQuery la consulta de búsqueda de la matriz
Tipo de Retorno	osid.logging.LogEntryList la LogEntryList retornada
Errores	NULL_ARGUMENT logEntryQuery es nulo OPERATION_FAILED incapaz de completar la solicitud

	PERMISSION_DENIED autorización es un fracaso UNSUPPORTED logEntryQuery no es de este servicio		
Método	getLogEntrySearch		
Descripción	Obtiene el log de entrada de la interfaz de búsqueda.		
Tipo de Retorno	osid.logging.LogEntrySearch el log de entrada de la interfaz de búsqueda.		
Método	getLogEntrySearchOrder		
Descripción	Obtiene un log de entrada para el orden de la interfaz de búsqueda. El LogEntrySearchOrder se suministra a una LogEntrySearch para especificar el orden de los resultados		
Tipo de Retorno	osid.logging.LogEntrySearchOrder el log de entrada para el orden de la interfaz de búsqueda		
Método	getLogEntriesBySearch		
Descripción	Recibe los resultados de la búsqueda correspondiente a la consulta de búsqueda utilizando la búsqueda.		
Parámetros	osid.logging.LogEntrySearch	logEntrySearch	La interfaz de búsqueda
	osid.logging.LogEntryQuery	logEntryQuery	La matriz de la consultad de búsqueda
Tipo de Retorno	osid.logging.LogEntrySearchResults los resultados de búsqueda retornados.		
Errores	NULL_ARGUMENT logEntryQuery or logEntrySearch es nulo OPERATION_FAILED incapaz de completar la solicitud PERMISSION_DENIED autorización es un fracaso UNSUPPORTED logEntryQuery or logEntrySearch no es de este servicio.		

Tabla 65. Métodos de la interfaz *osid.logging.LogEntrySearchSession*

Interfaz *osid.logging.LogEntryAdminSession*

Método	getLogEntriesBySearch		
Descripción	Obtiene los resultados de la búsqueda correspondiente a la consulta de búsqueda utilizando la búsqueda.		
Parámetros	osid.logging.LogEntrySearch	logEntrySearch	La interfaz de búsqueda
	osid.logging.LogEntryQuery	logEntryQuery	La matriz de consulta de búsqueda
Tipo de Retorno	osid.logging.LogEntrySearchResults el resultado de búsqueda retornado.		
Errores	NULL_ARGUMENT logEntryQuery or logEntrySearch es nulo OPERATION_FAILED incapaz de completar la solicitud PERMISSION_DENIED autorización es un fracaso UNSUPPORTED logEntryQuery or logEntrySearch no es de este servicio.		
Método	canDeleteLogEntries		
Descripción	Prueba si este usuario puede eliminar las log de entradas. El retorno verdadero no garantiza éxito en la autorización, falso indica que es conocido eliminando un LogEntry que resultarán en un PERMISSION_DENIED. Esto es como una pista para una aplicación que no desean ofrecer a eliminar las operaciones de usuarios no autorizados.		
Tipo de Retorno	Boolean falso si la supresión de LogEntry no es autorizada, sino verdadero.		
Método	canDeleteLogEntry		
Descripción	Prueba si un usuario puede eliminar un LogEntry especificado. El retorno verdadero no garantiza éxito en la autorización, falso indica que es conocido eliminando un LogEntry que resultarán en un PERMISSION_DENIED. Esto es como una pista para una aplicación que no desean ofrecer a eliminar las operaciones de usuarios no autorizados.		
Parámetros	osid.id.Id	logEntryId	el Id del LogEntry
Tipo de Retorno	boolean falso si la supresión de LogEntry no es autorizada, sino verdadero		
Errores	NULL_ARGUMENT logEntryId es nulo		

Método	deleteLogEntry		
Descripción	Elimina un LogEntry.		
Parámetros	osid.id.Id	logEntryId	el Id del logEntryId para eliminar
Errores	NOT_FOUND logEntryId no encontrado NULL_ARGUMENT logEntryId es nula OPERATION_FAILED incapaz de completar la solicitud PERMISSION_DENIED autorización es un fracaso		
Método	deleteLogEntriesByPriorityType		
Descripción	Elimina entradas de log basado en el tipo de prioridad. Todas las entradas con las prioridades igual o inferior a la facilitada prioridad se borran.		
Parámetros	osid.type.Type	priorityType	La prioridad de logging
Errores	NULL_ARGUMENT priorityType es nulo OPERATION_FAILED incapaz de completar la solicitud PERMISSION_DENIED autorización es un fracaso		
Método	deleteLogEntriesByTimestamp		
Descripción	Elimina entradas del log entre las fechas ofrecidas.		
Parámetros	osid.calendaring.DateTime	inicio	La hora de inicio
	osid.calendaring.DateTime	fin	La hora de fin
Errores	INVALID_ARGUMENT inicio es superior a fin OPERATION_FAILED incapaz de completar la solicitud NULL_ARGUMENT inicio o fin es nulo PERMISSION_DENIED autorización es un fracaso		
Método	deleteLogEntriesByPriorityAndTimestamp		
Descripción	Elimina entradas del log entre las fechas ofrecidas para un determinado tipo de prioridad.		
Parámetros	osid.type.Type	priorityType	Prioridad de logging

	osid.calendaring.DateTime	start	Hora de inicio
	osid.calendaring.DateTime	end	Hora de fin
Errores	INVALID_ARGUMENT inicio es superior a fin OPERATION_FAILED incapaz de completar la solicitud NULL_ARGUMENT priorityType, inicio o fin es nulo PERMISSION_DENIED autorización es un fracaso		
Método	addIdToLogEntry		
Descripción	Añade un Id a un logentry con el fin de crear la compatibilidad. El ID principal de logEntry está determinada por el proveedor. El nuevo Id funciona como un alias al Id primario.		
Parámetros	osid.id.Id	logEntryId	el Id de un LogEntry
	osid.id.Id	aliasId	el alias Id
Errores	ALREADY_EXISTS aliasId ya se ha asignado NOT_FOUND logEntryId no encontrado NULL_ARGUMENT logEntryId o aliasId es nulo OPERATION_FAILED incapaz de completar la solicitud PERMISSION_DENIED autorización es un fracaso		

Tabla 66. Métodos de la Interfaz *osid.logging.LogEntryAdminSession*

Interfaz *osid.logging.LogEntryNotificationSession*

Método	canRegisterForLogEntryNotifications
Descripción	Pruebas si este usuario puede registrar las notificaciones de un log de entrada. El retorno verdadero no garantiza el éxito de la autorización. El retorno falso indica que se conoce todos los métodos en esta sesión que resultará en un PERMISSION_DENIED. Esto es como una pista para una aplicación que pueden optar por no ofrecer operaciones de notificación.
Tipo de Retorno	Boolean falso si los métodos de notificación no son autorizados, sino verdadero.

Método	useFederatedLogView
Descripción	Federa la vista de los métodos en esta sesión. Una vista federada incluye las entradas de los padres logs en la jerarquía de un log.
Método	uselolatedLogView
Descripción	Aisla la vista de los métodos en esta sesión. Una vista aislada restringe las notificaciones de las entradas a este único log.
Método	registerForNewLogEntries
Descripción	Registro para la notificación de un nuevo log de entrada. LogEntryReceiver.newLogEntry() es invocado cuando un nuevo LogEntry es creado.
Errores	PERMISSION_DENIED autorización es un fracaso OPERATION_FAILED incapaz de completar la solicitud.
Método	registerForDeletedLogEntries
Descripción	Registros para la notificación de un log de entrada eliminado. LogEntryReceiver.deletedLogEntry() es invocado cuando un log de entrada es eliminado.
Errores	OPERATION_FAILED incapaz de completar la solicitud PERMISSION_DENIED autorización es un fracaso

Tabla 67. Métodos de la Interfaz `osid.logging.LogEntryNotificationSession`

Interfaz `osid.logging.LogLookupSession`

Método	canLookupLogs
Descripción	Pruebas si este usuario puede registrar las búsquedas de un log. El retorno verdadero no garantiza el éxito de la autorización. El retorno falso indica que se

	conoce todos los métodos en esta sesión que resultará en un PERMISSION_DENIED. Esto es como una pista para una aplicación que pueden optar por no ofrecer operaciones de búsqueda para usuarios autorizados.		
Tipo de Retorno	Boolean falso si los métodos de búsqueda nos son autorizados, sino verdadero.		
Método	useComparativeLogView		
Descripción	El retorno de los métodos de búsqueda pueden omitir o traducir elementos basados en esta sesión, tal como autorización y no resultan en un error. Esta vista es usada cuando se desea una mayor interoperabilidad a expensas de precisión.		
Método	usePlenaryLogView		
Descripción	Una visión completa del retorno del log es deseado. Los métodos de retorno son solicitados o resulta en un error. Esta vista es usada cuando se desea mayor precisión en detrimento de la interoperabilidad.		
Método	getLog		
Descripción	Obtiene el log especificado por este Id. En modo plenario, el Id exacto se encuentra o es resuelto en un NOT_FOUND. En otro caso el log retornado puede tener un Id diferente que el requerido, tal como el caso donde un Id duplicado fue asignado a un log y retenido por compatibilidad.		
Parámetros	osid.id.Id	logId	Id del Log
Tipo de Retorno	osid.logging.Log de log		
Errores	NOT_FOUND logId no encontrado NULL_ARGUMENT logId es nulo OPERATION_FAILED incapaz de completar la solicitud PERMISSION_DENIED autorización es un fracaso		
Método	getLogsByIds		
Descripción	Obtiene una LogList correspondiente al IdList dado. En plenaria la lista retornada		

	contiene todos los log especificados en la lista Id, en el orden de la lista, incluyendo duplicados, o un error si los resultados en una ID de la lista facilitada no se encuentra o es inaccesibles. En otro caso, los logs inaccesibles pueden ser omitidos de la lista y puede presentar los elementos incluidos en el orden de retorno de un único conjunto.		
Parámetros	osid.id.IdList	logIdList	La lista de Ids recuperado
Tipo de Retorno	osid.logging.LogList la lista de log retornada		
Errores	NOT_FOUND un ID no fue encontrada NULL_ARGUMENT logIdList es nulo OPERATION_FAILED incapaz de completar la solicitud PERMISSION_DENIED autorización es un fracaso		
Método			
	getLogsByGenusType		
Descripción	Obtiene un LogList correspondiente al tipo del género del log dado que no incluye los tipos géneros del log derivado de un tipo especificado. En plenaria la lista retornada contiene todos los log conocidos o un resultado error. En otro caso, la lista retornada puede contener solo aquellos log que son accesibles mediante esta sesión. En ambos casos el orden de los conjuntos no es especificado.		
Parámetros	osid.type.Type	logGenusType	Un tipo de género del log
Tipo de Retorno	osid.logging.LogList the returned Log list		
Errores	NULL_ARGUMENT logGenusType es nulo Errors OPERATION_FAILED incapaz de completar la solicitud PERMISSION_DENIED autorización es un fracaso		
Método			
	getLogsByParentGenusType		
Descripción	Obtiene un LogList correspondiente al tipo del género del log dado e incluye cualquier log adicional con los tipos de géneros del log derivado de un tipo especificado. En plenaria, la lista retornada contiene todos los log conocidos o un resultado error. En otro caso la lista retornada puede contener solo aquellos log que		

	son accesibles mediante esta sesión. En ambos casos el orden de los conjuntos no es especificado.		
Parámetros	osid.type.Type	logGenusType	Un tipo de género del log
Tipo de Retorno	osid.logging.LogList la lista log retornada		
Errores	NULL_ARGUMENT logGenusType es nulo OPERATION_FAILED incapaz de completar la solicitud PERMISSION_DENIED autorización es un fracaso		
Método			
	getLogsByRecordType		
Descripción	Obtiene un LogList correspondiente al tipo de registro de un repositorio dado. En plenaria, la lista retornada contiene todos los log conocidos o un resultado error. En otro caso, la lista retornada puede contener solo aquellos log que son accesibles mediante esta sesión. En ambos casos el orden de los conjuntos no es especificado.		
Parámetros	osid.type.Type	logRecordType	Un tipo de registro de un log
Tipo de Retorno	osid.logging.LogList la lista log retornada		
Errores	NULL_ARGUMENT logRecordType es nulo OPERATION_FAILED incapaz de completar la solicitud PERMISSION_DENIED autorización es un fracaso		
Método			
	getLogs		
Descripción	Obtiene todos los logs. En plenaria, la lista retornada contiene todos los log conocidos o un resultado error. En otro caso la lista retornada puede contener solo aquellos log que son accesibles mediante esta sesión. En ambos casos el orden de los conjuntos no es especificado.		
Tipo de Retorno	osid.logging.LogList una lista de logs		
Errores	OPERATION_FAILED incapaz de completar la solicitud PERMISSION_DENIED autorización es un fracaso		

Tabla 68. Métodos de la Interfaz osid.logging.LogLookupSession

Interfaz osid.logging.LogSearchSession

Método	canSearchLogs		
Descripción	Pruebas si este usuario puede registrar las búsquedas de un log. El retorno verdadero no garantiza el éxito de la autorización. El retorno falso indica que se conoce todos los métodos en esta sesión que resultará en un PERMISSION_DENIED. Esto es como una pista para una aplicación que pueden optar por no ofrecer operaciones de búsqueda para un usuario no autorizado.		
Tipo de Retorno	boolean falso si el método de búsqueda no es autorizado, sino verdadero		
Método	getLogQuery		
Descripción	Obtiene un log de la interfaz de búsqueda. La consulta retornada no tendrá un consulta de extensión		
Tipo de Retorno	osid.logging.LogQuery la consulta de log		
Método	getLogsByQuery		
Descripción	Obtiene una lista de logs para un ainterfaz de búsqueda dada.		
Parámetros	osid.logging.LogQuery	logQuery	La matriz de consulta de búsqueda
Tipo de Retorno	osid.logging.LogList retornada la LogList		
Errores	NULL_ARGUMENT logQuery es nulo OPERATION_FAILED incapaz de completar la solicitud PERMISSION_DENIED autorización es un fracaso UNSUPPORTED logQuery no es de este servicio		
Método	getLogSearch		
Descripción	Obtiene un log de interfaz de búsqueda		
Tipo de Retorno	osid.logging.LogSearch la interfaz de búsqueda de un log		

Método	getLogSearchOrder		
Descripción	Obtiene un log de la interfaz de búsqueda ordenada. El LogSearchOrder es suministrado a un LogSearch para especificar el orden de los resultados.		
Tipo de Retorno	osid.logging.LogSearchOrder la interfaz de búsqueda ordenada de un log		
<hr/>			
Método	getLogsBySearch		
Descripción	Obtiene los resultados de búsqueda de una interfaz de búsqueda dada.		
Parámetros	osid.logging.LogQuery	logQuery	La matriz de la consulta de búsqueda
	osid.logging.LogSearch	logSearch	La interfaz de búsqueda
Tipo de Retorno	osid.logging.LogSearchResults el resultado de búsqueda		
Errores	NULL_ARGUMENT logQuery o logSearch es nulo OPERATION_FAILED incapaz de completar la solicitud PERMISSION_DENIED autorización es un fracaso UNSUPPORTED logQuery o logSearch no es de este servicio		

Tabla 69. Métodos de la Interfaz *osid.logging.LogSearchSession*

Interfaz *osid.logging.LogAdminSession*

Método	canCreateLogs		
Descripción	Pruebas si este usuario puede logs. El retorno verdadero no garantiza el éxito de la autorización. El retorno falso indica que se conoce la creación de un log que resultará en un PERMISSION_DENIED. Esto es como una pista para una aplicación que pueden optar por no ofrecer operaciones de crear para un usuario no autorizado.		
Tipo de Retorno	boolean falso si la creación del log no es autorizada, sino verdadero		
<hr/>			
Método	canCreateLogWithRecordTypes		
Descripción	Pruebas si este usuario puede crear un único log usando los tipos de interfaz de registros deseados. Mientras que LoggingManager.getLogRecordTypes() puede ser		

	usado para examinar cuales de las interfaces de registro son soportadas, este método prueba cual registro es requerido para la creación de un log específico. Proporcionando una entrada de matriz que prueba si un log puede ser creado con ningún registro.		
Parámetros	osid.type.Type[]	logRecordTypes	Tipo de matriz
Tipo de Retorno	Boolean, verdadero si es soportada la creación de un log usando un tipo de registro especificado, sino falso.		
Errores	NULL_ARGUMENT logRecordTypes es nulo		
Método			
Método	getLogFormForCreate		
Descripción	Obtiene el formulario de un log para la creación de un nuevo log. Un nuevo formulario debe ser solicitado por cada transacción creada.		
Tipo de Retorno	osid.logging.LogForm el formulario del log		
Método			
Método	createLog		
Descripción	Crear un nuevo log.		
Parámetros	osid.logging.LogForm logForm el formulario para este Log		
Tipo de Retorno	osid.logging.Log el Nuevo log		
Errores	ALREADY_EXISTS intento de duplicación de una propiedad es el sistema subyacente a la aplicación de ser único INVALID_ARGUMENT uno o más de los elementos de formulario no es válido NULL_ARGUMENT logForm es nulo OPERATION_FAILED incapaz de completar la solicitud PERMISSION_DENIED autorización es un fracaso UNSUPPORTED logForm no es de este servicio		
Método			
Método	canUpdateLogs		
Descripción	Pruebas si este usuario puede actualizar un log. El retorno verdadero no garantiza el éxito de la autorización. El retorno falso indica que se conoce el log actualizado		

	que resultará en un PERMISSION_DENIED. Esto es como una pista para una aplicación que pueden optar por no ofrecer operaciones de actualización para un usuario no autorizado.		
Tipo de Retorno	Boolean falso si la modificación del log no es autorizada, sino false.		
Método			
	canUpdateLog		
Descripción	Pruebas si este usuario puede actualizar un log especificado. El retorno verdadero no garantiza el éxito de la autorización. El retorno falso indica que se conoce el log actualizado que resultará en un PERMISSION_DENIED. Esto es como una pista para una aplicación que pueden optar por no ofrecer operaciones de actualización para un usuario no autorizado.		
Parámetros	osid.id.Id	logId	el Id del Log
Tipo de Retorno	boolean falso si la modificación del log no es autorizado, sino verdadero		
Errores	NULL_ARGUMENT logId es nulo		
Método			
	getLogFormForUpdate		
Descripción	Obtiene el formulario de un log para actualizar un log existente. Un nuevo formulario de un log debe ser solicitado para cada transacción de actualización.		
Parámetros	osid.id.Id	logId	el Id del Log
Tipo de Retorno	osid.logging.LogForm el formulario del log		
Errores	NULL_ARGUMENT logId es nulo		
Método			
	updateLog		
Descripción	Actualizar un log existente		
Parámetros	osid.id.Id	logId	el Id del Log
	osid.logging.LogForm	logForm	El formulario que contiene los elemntos a ser actualizados
Errores	INVALID_ARGUMENT el formulario contiene valosre inválidos. NOT_FOUND logId no encontrado		

	NULL_ARGUMENT logId o logForm es nulo OPERATION_FAILED incapaz de completar la solicitud PERMISSION_DENIED autorización es un fracaso UNSUPPORTED logForm no es compatible		
Método	canDeleteLogs		
Descripción	Pruebas si este usuario puede eliminar un log. El retorno verdadero no garantiza el éxito de la autorización. El retorno falso indica que se conoce el log eliminado que resultará en un PERMISSION_DENIED. Esto es como una pista para una aplicación que pueden optar por no ofrecer operaciones de eliminar para un usuario no autorizado.		
Tipo de Retorno	boolean falso si la eliminación de un Log no es autorizado, sino verdadero		
Método	canDeleteLog		
Descripción	Pruebas si este usuario puede eliminar un log especificado. El retorno verdadero no garantiza el éxito de la autorización. El retorno falso indica que se conoce el log eliminado que resultará en un PERMISSION_DENIED. Esto es como una pista para una aplicación que pueden optar por no ofrecer operaciones de eliminar para un usuario no autorizado.		
Parámetros	osid.id.Id	logId	el Id del Log
Tipo de Retorno	boolean falso si la eliminación del Log no es autorizado, sino verdadero		
Errores	NULL_ARGUMENT logId es nulo		
Método	deleteLog		
Descripción	Eliminar un log		
Parámetros	osid.id.Id	logId	el Id del Log a eliminar
Errores	NOT_FOUND logId no encontrado NULL_ARGUMENT logId es nulo OPERATION_FAILED incapaz de completar la solicitud		

	PERMISSION_DENIED autorización es un fracaso		
Método	addIdToLog		
Descripción	Añade a un ID de un log con el fin de crear la compatibilidad. El ID principal del log es determinado por el proveedor. El nuevo Id funciona como un alias al Id primario.		
Parámetros	osid.id.Id	logId	el Id de un Log
	osid.id.Id	aliasId	el alias Id
Errores	ALREADY_EXISTS aliasId ya se ha asignado NOT_FOUND logId no encontrado NULL_ARGUMENT logId or aliasId es nulo OPERATION_FAILED incapaz de completar la solicitud PERMISSION_DENIED autorización es un fracaso		

Tabla 70. Métodos de la Interfaz `osid.logging.LogAdminSession`

Interfaz `osid.logging.LogNotificationSession`

Método	canRegisterForLogNotifications
Descripción	Pruebas si este usuario puede registrar las notificaciones para un log. El retorno verdadero no garantiza el éxito de la autorización. El retorno falso indica que se conoce todos los métodos en esta sesión que resultará en un PERMISSION_DENIED. Esto es como una pista para una aplicación que pueden optar por no ofrecer operaciones de notificación.
Tipo de Retorno	boolean falso si los métodos de notificación no son autorizados, sino verdadero
Método	registerForNewLogs
Descripción	Registro para las notificaciones de los nuevos logs. <code>LogReceiver.newLog ()</code> se invoca cuando un nuevo log es creado
Errores	OPERATION_FAILED incapaz de completar la solicitud PERMISSION_DENIED autorización es un fracaso

Método	registerForNewLogAncestors		
Descripción	Registros para la notificación de una estructura de jerarquía actualizada que introduce un nuevo antecesor del log especificado. LogReceiver.newAncestorLog () se invoca cuando el log especificado obtiene el nodo a un nuevo antecesor.		
Parámetros	osid.id.Id	logId	el Id del nodo del Log a vigilar
Errores	NOT_FOUND un nodo de un log no fue identificado por el Id dado NULL_ARGUMENT logId es nulo OPERATION_FAILED incapaz de completar la solicitud PERMISSION_DENIED autorización es un fracaso		
Método	registerForNewLogDescendants		
Descripción	Registros para la notificación de una estructura de jerarquía actualizada que introduce un nuevo descendiente del log especificado. LogReceiver.newDescendantLog () se invoca cuando el log especificado obtiene el nodo a un nuevo descendiente.		
Parámetros	osid.id.Id	logId	el Id del nodo del Log a vigilar
Errores	NOT_FOUND un nodo de un log no fue identificado por el Id dado NULL_ARGUMENT logId es nulo OPERATION_FAILED incapaz de completar la solicitud PERMISSION_DENIED autorización es un fracaso		
Método	registerForChangedLogs		
Descripción	Refistro por la notificación de un log actualizado. LogReceiver.changedLog() es invocado cuando un log es cambiado.		
Errores	OPERATION_FAILED incapaz de completar la solicitud PERMISSION_DENIED autorización es un fracaso		
Método	registerForChangedLog		

Descripción	Refistro por la notificación de un log actualizado. LogReceiver.changedLog() es invocado cuando un log especificado es cambiado		
Parámetros	osid.id.Id	logId	el Id del Log a vigilar
Errores	NOT_FOUND un log no fue identificado por el Id dado NULL_ARGUMENT logId es nulo OPERATION_FAILED incapaz de completar la solicitud PERMISSION_DENIED autorización es un fracaso		
Método	registerForDeletedLogs		
Descripción	Refistro por la notificación de un log eliminado. LogReceiver.deletedLog() es invocado cuando un log es eliminado.		
Errores	OPERATION_FAILED incapaz de completar la solicitud PERMISSION_DENIED autorización es un fracaso		
Método	registerForDeletedLog		
Descripción	Refistro por la notificación de un log eliminado. LogReceiver.deletedLog() es invocado cuando un log especificado es eliminado.		
Parámetros	osid.id.Id	logId	el Id del Log a vigilar
Errores	NOT_FOUND un log no fue identificado por el Id dado NULL_ARGUMENT logId es nulo OPERATION_FAILED incapaz de completar la solicitud PERMISSION_DENIED autorización es un fracaso		
Método	registerForDeletedLogAncestors		
Descripción	Registros para la notificación de una estructura de jerarquía actualizada que elimine un antecesor del log especificado. LogReceiver.deletedAncestor() se invoca cuando el log especificado pierde el nodo a un antecesor.		

Parámetros	osid.id.Id	logId	el Id del nodo del Log a vigilar
Errores	NOT_FOUND un nodo de un log no fue identificado por el Id dado NULL_ARGUMENT logId es nulo OPERATION_FAILED incapaz de completar la solicitud PERMISSION_DENIED autorización es un fracaso		
Método	registerForDeletedLogDescendants		
Descripción	Registros para la notificación de una estructura de jerarquía actualizada que elimine un descendiente del log especificado. LogReceiver.deletedDescendant() se invoca cuando el log especificado pierde el nodo a un descendiente.		
Parámetros	osid.id.Id	logId	el Id del nodo del Log a vigilar
Errores	NOT_FOUND un nodo de un log no fue identificado por el Id dado NULL_ARGUMENT logId es nulo OPERATION_FAILED incapaz de completar la solicitud PERMISSION_DENIED autorización es un fracaso		

Tabla 71. Métodos de la Interfaz `osid.logging.LogNotificationSession`

Interfaz `osid.logging.LogEntryQuery`

Método	matchPriority		
Descripción	Coincide con un tipo de prioridad para la entrada de log.		
Parámetros	osid.type.Type	priorityType	Type de partido
	boolean	match	verdadero si es por el partido positivo, falso para el partido negativo
Errores	NULL_ARGUMENT priorityType es nulo		
Método	matchTimestamp		
Descripción	Coincide con este tipo de prioridad para este log de entrada		
Parámetros	osid.calendar.DateTime	startTime	Tiempo de inicio

	osid.calendaring.DateTime	endTime	Tiempo de fin
	boolean	match	verdadero si es por el partido positivo, falso para el partido negativo
Errores	INVALID_ARGUMENT el tiempo de inicio es mayor que el de fin NULL_ARGUMENT startTime o endTime es nulo		
Método			
	matchAgentId		
Descripción	Coincide con un agente en este log de entrada.		
Parámetros	osid.id.Id	agentId	Id del partido
	boolean	match	verdadero si es partido positivo, falso para el partido negativo
Errores	NULL_ARGUMENT agentId es nulo		
Método			
	supportsAgentQuery		
Descripción	Pruebas si un AgentQuery está disponible para consulta de los agentes.		
Tipo de Retorno	boolean verdadero si un agente de interfaz de consulta es disponible, sino falso		
Método			
	getAgentQuery		
Descripción	Obtiene la interfaz de consulta para un agente		
Tipo de Retorno	osid.authentication.AgentQuery la consulta del agente		
Errores	UNIMPLEMENTED supportsAgentQuery() es falso		
Método			
	matchContentType		
Descripción	Coincide con el tipo de contenido para este log de entrada.		
Parámetros	osid.type.Type	contentType	Tipo de partido
	boolean	match	verdadero si es partido positivo, falso para el partido negativo
Errores	NULL_ARGUMENT contentType es nulo		

Método	supportsContentQuery
Descripción	Prueba si el contenido de la consulta esta disponible
Tipo de Retorno	boolean verdadero si el contenido de consulta es disponible, sino falso
<hr/>	
Método	getContentQuery
Descripción	Obtiene el contenido de la consulta correspondiente al tipo de interfaz de contenido solicitada en esta sesión de búsqueda.
Tipo de Retorno	Objeto de contenido de consulta
Errores	UNIMPLEMENTED supportsContentQuery() es falso

Tabla 72. Métodos de la Interfaz *osid.logging.LogEntryQuery*

Interfaz *osid.logging.LogEntrySearchOrder*

Método	orderByPriorityType
Descripción	Especifica una preferencia para los logs de entrada ordenándolos por un tipo de prioridad.
<hr/>	
Método	orderByTime
Descripción	Especifica una preferencia para los logs de entrada ordenándolos por el tiempo
<hr/>	
Método	orderByAgent
Descripción	Especifica una preferencia para los logs de entrada ordenándolos por agente
<hr/>	
Método	orderByContentType
Descripción	Especifica una preferencia para los logs de entrada ordenándolos por el tipo de contenido

Tabla 73. Métodos de la Interfaz *osid.logging.LogEntrySearchOrder*

Interfaz *osid.logging.LogEntryReceiver*

Método	newLogEntry		
Descripción	La devolución de llamada para las notificaciones de nuevas entradas del log.		
Parámetros	osid.id.Id	registeredLogId	el Id del log registrado del Id del log que dio origen a este evento
	osid.id.Id	logId	la misma del log registrado o una de sus descendiente
	osid.id.Id	entryId	el Id de la nueva entrada.
Método	deletedLogEntry		
Descripción	La llamada de notificación de la entrada del log eliminado.		
Parámetros	osid.id.Id	registeredLogId	el Id del log registrado del Id del log que dio origen a este evento
	osid.id.Id	logId	la misma del log registrado o una de sus descendiente
	osid.id.Id	entryId	el Id de la entrada eliminada

Tabla 74. Métodos de la Interfaz `osid.logging.LogEntryReceiver`

Interfaz `osid.logging.LogEntryList`

Método	getNextLogEntry
Descripción	Obtiene el proximo LogEntry en esta lista.El hasNext() del proximo LogEntry en esta lista
Tipo de Retorno	osid.logging.LogEntry método que debe ser utilizado para probar la proxima LogEntry esta disponible antes de la llamada a este método
Errores	ILLEGAL_STATE no tiene elementos disponibles en esta lista OPERATION_FAILED incapaz de completar la solicitud
Método	getNextLogEntries
Descripción	Obtiene el proximo conjunto de elementos LogEntry en esta lista.La cantidad especificada debe ser menor o igual a lo retornado por available().

Parámetros	cardinal	n	El numero de elementos LogEntry solicitados los cuales deben ser menores o iguales a available()
Tipo de Retorno	osid.logging.LogEntry un arreglo de elementos LogEntry, la longitud de del arreglo es menor o igual que el numero especificado		
Errores	ILLEGAL_STATE no tiene elementos disponibles en esta lista OPERATION_FAILED incapaz de completar la solicitud		

Tabla 75. Métodos de la interfaz `osid.logging.LogEntryList`

Interfaz `osid.logging.LogQuery`

Método	<code>supportsLogEntryQuery</code>		
Descripción	Prueba si una consulta de un log de entrada está disponible		
Tipo de Retorno	boolean verdadero si la consulta del log de entrada esta disponible, sino falso		
<hr/>			
Método	<code>getLogEntryQuery</code>		
Descripción	Obtiene la interfaz de consulta para un log de entrada		
Tipo de Retorno	<code>osid.logging.LogEntryQuery</code> la consulta del log de entrada		
Errores	UNIMPLEMENTED <code>supportsLogEntryQuery()</code> es falso		
<hr/>			
Método	<code>getLogQueryRecord</code>		
Descripción	Obtiene la interfaz de registro de consulta correspondiente al tipo de log de registro. Múltiples recuperaciones de registros producen un booleano anidado o termino		
Parámetros	<code>osid.type.Type</code>	<code>logRecordType</code>	Un tipo de registro log
Tipo de Retorno	<code>osid.logging.LogQueryRecord</code> el registro de consulta log		
Errores	NULL_ARGUMENT <code>logRecordType</code> es nulo OPERATION_FAILED incapaz de completar la solicitud PERMISSION_DENIED autorización ocurrida un fracaso UNSUPPORTED <code>hasRecordType(logRecordType)</code> es falso		

Tabla 76. Métodos de la Interfaz `osid.logging.LogQuery`

Interfaz osid.logging.LogForm

Método	getLogFormRecord		
Descripción	Obtiene la interfaz LogFormRecord correspondiente al tipo de interfaz de log de registro dado		
Parámetros	osid.type.Type	logRecordType	El tipo de registro log
Tipo de Retorno	osid.logging.LogFormRecord el registro		
Errores	NULL_ARGUMENT logRecordType es nulo OPERATION_FAILED incapaz de completar la solicitud PERMISSION_DENIED autorización ocurrida un fracaso UNSUPPORTED hasRecordType(logRecordType) es falso		

Tabla 77. Métodos de la Interfaz osid.logging.LogForm

Interfaz osid.logging.LogSearchOrder

Método	getLogSearchOrderRecord		
Descripción	Obtiene el orden de registro de log de búsqueda correspondiente al log tipo de registro dado. Múltiples recuperaciones devuelven los mismos objetos resaltados		
Parámetros	osid.type.Type	logRecordType	Un tipo de registro log
Tipo de Retorno	osid.logging.LogSearchOrderRecord la interfaz de registro ordenada del log de búsqueda		
Errores	NULL_ARGUMENT logRecordType es nulo OPERATION_FAILED incapaz de completar la solicitud PERMISSION_DENIED autorización ocurrida un fracaso UNSUPPORTED hasRecordType(logRecordType) es falso		

Tabla 78. Métodos de la Interfaz osid.logging.LogSearchOrder

Interfaz osid.logging.LogSearch

Método	searchWithinLogResults		
Descripción	Ejecución de búsqueda usando un resultado de búsqueda previo		
Parámetros	osid.logging.LogSearchResults	results	los resultados de una consulta
Errores	INVALID_ARGUMENT el resultado no es válido NULL_ARGUMENT resultado es nulo		
<hr/>			
Método	searchAmongLogs		
Descripción	Ejecutar esta búsqueda entre los registros de la lista dada.		
Parámetros	osid.id.IdList	logIds	Listas de logs
Errores	NULL_ARGUMENT logIds es nulo		
<hr/>			
Método	orderLogResults		
Descripción	Especificadamente una ordenamiento de los resultados de búsqueda.		
Parámetros	osid.logging.LogSearchOrder	ogSearchOrder	El log de búsqueda ordenado
Errores	NULL_ARGUMENT logSearchOrder es nulo UNSUPPORTED logSearchOrder no es de este servicio		
<hr/>			
Método	getLogSearchRecord		
Descripción	Obtiene el registro correspondiente al log tipo de registro de búsqueda dado. Este método debe ser utilizado para recuperar un objeto implementando la interfaz de registro solicitada a lo largo de sus interfaces ancestros.		
Parámetros	osid.type.Type	logSearchRecordType	Un tipo de registro de un log de búsqueda
Tipo de Retorno	osid.logging.LogSearchRecord la interfaz de búsqueda del log		
Errores	NULL_ARGUMENT logSearchRecordType es nulo OPERATION_FAILED incapaz de completar la solicitud PERMISSION_DENIED autorización ocurrida un fracaso UNSUPPORTED hasSearchRecordType(logSearchRecordType) es falso		

Tabla 79. Métodos de la Interfaz *osid.logging.LogSearch***Interfaz *osid.logging.LogSearchResults***

Método	getLogs		
Descripción	Obtiene el registro de la lista resultante de una búsqueda.		
Tipo de Retorno	osid.logging.LogList la lista de log		
Método			
Método	getLogSearchResultsRecord		
Descripción	Obtiene el registro correspondiente al log tipo de registro de búsqueda dado. Este método debe ser utilizado para recuperar un objeto implementando la interfaz de registro solicitado a lo largo de todos sus interfaces ancestros		
Parámetros	osid.type.Type	logSearchRecordType	Un tipo de registro de un log de búsqueda
Tipo de Retorno	osid.logging.LogSearchResultsRecord la interfaz de búsqueda del log		
Errores	NULL_ARGUMENT logSearchRecordType es nulo OPERATION_FAILED incapaz de completar la solicitud PERMISSION_DENIED autorización ocurrida un fracaso UNSUPPORTED hasSearchRecordType(logSearchRecordType) es falso		

Tabla 80. Métodos de la Interfaz *osid.logging.LogSearchResults***Interfaz *osid.logging.LogReceiver***

Método	newLog		
Descripción	La llamada de notificación de un Nuevo log		
Parámetros	osid.id.Id	logId	el Id del Nuevo log
Método			
Método	newAncestorLog		
Descripción	La llamada para las notificaciones de los antecesores de un nuevo log.		
Parámetros	osid.id.Id	logId	El Id del log registrado

	osid.id.Id	ancestorId	el Id del Nuevo log antecesor
Método	newDescendantLog		
Descripción	La llamada por notificación de un Nuevo descendiente de un log		
Parámetros	osid.id.Id	logId	el Id de un log registrado
	osid.id.Id	descendantId	el Id de un nuevo log descendiente
Método	changedLog		
Descripción	La llamada por notificación de un log actualizado		
Parámetros	osid.id.Id	logId	el Id del log actualizado
Método	deletedAncestorLog		
Descripción	La llamada por notificación de los ancestors eliminados de un log		
Parámetros	osid.id.Id	logId	el Id de un log registrado
	osid.id.Id	ancestorId	el Id del ancestro eliminado del log
Método	deletedDescendantLog		
Descripción	La llamada por notificación del descendiente eliminado de un log		
Parámetros	osid.id.Id	logId	el Id de un log registrado
	osid.id.Id	descendantId	el Id del descendiente eliminado del log

Tabla 81. Métodos de la interfaz *osid.logging.LogReceiver*

Anexo 6. Otros métodos del estándar IMS ES

Resumen de las operaciones PersonManagerService

Método	Descripción
createPersons	Para solicitar la creación de un conjunto poblado del registro 'persona' en el sistema de destino. La fuente es la responsable de la asignación de los identificadores únicos.
createByProxyPersons	Para solicitar la creación de un conjunto de los registros de 'persona' poblados en el sistema de destino. El objetivo es responsable de la asignación de los identificadores únicos.
deletePersons	Para solicitar la eliminación de un conjunto de registro «persona». El registro «persona» y todas sus relaciones son eliminados.
readPersons	Para leer todo el contenido del conjunto de los registros de 'persona' identificados. El objetivo debe devolver todos los datos que tiene para cada un registro de 'persona' identificado.
readPersonsForGroup	Para recuperar los registros "persona" para un determinado grupo. Esto devuelve el registro de cada persona que es miembro del Grupo, es decir, para quienes un registro de miembros en el Grupo existe.
updatePersons	Para escribir nuevos contenidos en el conjunto de registros 'persona' identificados. El objetivo debe escribir los nuevos datos en cada uno de los registros 'persona'. Estas operaciones son aditivas.
replacePersons	Para reemplazar el contenido de un conjunto de registros 'persona' identificados. El objetivo debe escribir los nuevos datos en los registros 'persona'.
changePersonsIdentifier	Para cambiar el Id buscado en un conjunto de registros 'persona'. La finalización de esta operación se traducirá en acciones retrasadas usando el original de presentación de informes sourcedId un identificador de estatus desconocido.

Tabla 82. Resumen de las operaciones PersonManagerService

Resumen de las operaciones GroupManagerService

Método	Descripción
createGroups	Para solicitar la creación de un conjunto de los registros de Grupo en el sistema de destino y la fuente es la responsable de la asignación de cada uno de los identificadores únicos.
createByProxyGroups	Para solicitar la creación de un conjunto de registros Grupo en el sistema de destino. Es responsable de la asignación de los identificadores únicos.
deleteGroups	Para solicitar la eliminación de un conjunto de registro de «grupo». Los registros de «grupo» y todas sus relaciones asociadas se borran.
deleteGroupsRelationship	Para solicitar la eliminación de un conjunto de relaciones en el seno de los registros de los Grupo. Esto no elimina los correspondientes registros de los Grupo.
readGroups	Para leer todo el contenido del conjunto de registros de «grupo». El objetivo debe devolver todos los datos que tiene para cada uno de los registros de los Grupo.
readGroupsForPerson	Para recuperar los registros de 'grupo' por una persona en particular.
updateGroups	Para escribir nuevos contenidos en el conjunto de los registros de «grupo». El objetivo debe escribir los nuevos datos en cada uno de los registros de Grupo
replaceGroups	Para reemplazar el contenido de un conjunto de los registros de Grupo. El objetivo debe escribir los nuevos datos en los registros del Grupo.
changeGroupsIdentifier	Para cambiar el sourcedId del registro de «grupo». La finalización de esta operación se traducirá en acciones retrasadas usando la presentación original de informes sourcedId un identificador de estatus desconocido.

Tabla 83. Resumen de las operaciones GroupManagerService

Resumen de las operaciones MembershipManagerService

Método	Descripción
createMemberships	Para solicitar la creación de un conjunto de registros "miembro" en el sistema de destino. La fuente es la responsable de la asignación de los identificadores únicos.
createByProxyMemberships	Para solicitar la creación de un conjunto de registros "miembro" en el sistema de destino y el objetivo es responsable de la asignación de cada uno de los identificadores únicos.
deleteMemberships	Para solicitar la eliminación de un conjunto de registro de "miembro". Los registros "miembro" y todos sus relaciones se borran.
readMemberships	Para leer todo el contenido del conjunto de registros de "miembro". El objetivo debe devolver todos los datos que tiene para cada uno de los registros "miembro" identificados.
readMembershipsForPerson	Para recuperar todos los registros "miembros" para una persona en particular. Esto devuelve todos los registros 'miembros' para el registro 'persona' identificado.
readMembershipsForGroup	Para recuperar todos los registros 'miembros' para un determinado grupo. Esto devuelve todos los registros 'miembros' para el registro «grupo» identificado.
updateMemberships	Para escribir nuevos contenidos en el conjunto de los registros de "miembro". El objetivo debe escribir los nuevos datos en cada uno de los registros 'miembros'.

Tabla 84. Resumen de las operaciones MembershipManagerService

GLOSARIO DE TÉRMINOS

C

Componente: Piezas funcionales de un conjunto integrado que sería las Universidades Virtuales.

E

E-learning: conjunto de tecnologías, aplicaciones y servicios orientados a facilitar la enseñanza y el aprendizaje a través de Internet/Intranet, que facilitan el acceso a la información y la comunicación con otros participantes. (Red TTnet, 2005)

Estándar: es un patrón, una tipificación o una norma de cómo realizar algo (AulaGlobal, 2007).

I

Interoperabilidad: la habilidad de dos o más sistemas o componentes para intercambiar información y para usar la información que ha sido intercambiada.(IEEE, 2007)

M

Modelo: Un modelo identifica elementos importantes de un contexto o una realidad y establece relaciones entre éstos con el fin de explicar dicha realidad.

U

Universidad Virtual: Es una forma flexible de enseñanza especializada a distancia que utiliza avanzadas tecnologías de comunicación y genera nuevos espacios para que la comunidad universitaria pueda llevar a cabo sus procesos de comunicación, gestión, servicios e investigación.