



# Universidad Austral de Chile

Facultad de Ciencias de la Ingeniería

Escuela de Ingeniería Civil en Informática

## IMPLEMENTACIÓN DE UN REPOSITORIO DIGITAL EMPLEANDO TECNOLOGÍAS DE CÓDIGO ABIERTO

Tesis para optar al Título de:  
Ingeniero Civil en Informática

Profesor Patrocinante:  
Sr. Luis Vidal Vidal  
Ingeniero Civil en Informática  
Licenciado en Ciencias de la Ingeniería

Profesor Co-Patrocinante:  
Sr. Mauricio Ruiz-Tagle Molina  
Ingeniero Civil en Informática  
D.E.A. en Informática

Profesor Informante:  
Sr. Luis Alvares González  
Ingeniero Civil Electricista  
Magíster en Ingeniería Informática  
D.E.A Técnicas Informáticas Avanzadas

GERARDO ANDRES MARTÍNEZ ROSAS  
VALDIVIA – CHILE  
2009



Valdivia, 13 de Enero de 2009.

De : Luis Hernán Vidal Vidal.  
A : Sr. Juan Pablo Salazar F.  
Director de Escuela de Ingeniería Civil en Informática.  
Ref. : Informa Calificación Trabajo de Titulación.

MOTIVO: Informar revisión y calificación del Proyecto de Título "Implementación de un repositorio digital empleando tecnologías de código abierto.", presentado por el alumno GERARDO ANDRÉS MARTÍNEZ ROSAS, que refleja lo siguiente:

Se logró el objetivo planteado que permitió Implementar un Repositorio Digital, utilizando como componentes principales el Gestor Documental FEDORA (back-end) y una Interfaz Web (front-end), con el objeto de cooperar en el perfeccionamiento del Proyecto Fedora y generar pautas que permitan la implementación de un Repositorio Digital en una organización.

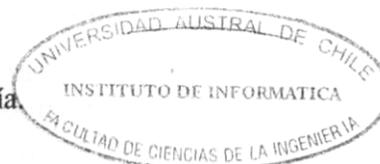
El esfuerzo constante, junto con la dedicación al tema son aspectos destacables durante el proceso del trabajo realizado.

Cumplimiento del objetivo propuesto.	7,0
Satisfacción de alguna necesidad.	7,0
Aplicación del método científico.	6,5
Interpretación de los datos y obtención de conclusiones.	6,7
Originalidad.	6,7
Aplicación de criterios de análisis y diseño.	6,7
Perspectivas del trabajo.	7,0
Coherencia y rigurosidad lógica.	7,0
Precisión del lenguaje técnico en la exposición, composición, redacción e ilustración.	7,0
Evaluación Tesis.	6,8

Por todo lo anterior expuesto califico el trabajo de titulación del Sr. GERARDO ANDRÉS MARTÍNEZ ROSAS con nota 6,8 (seis coma ocho).

Sin otro particular, se despide atentamente.

Ing. Luis Hernán Vidal Vidal.  
Profesor Instituto de Informática.  
Facultad de Ciencias de la Ingeniería  
Universidad Austral de Chile.



Valdivia, 19 de enero de 2009

De: Mauricio Ruiz-Tagle Molina  
Co-Patrocinante del Proyecto de Título

A: Juan Pablo Salazar Fernández  
Director de la Escuela de Ingeniería Civil en Informática

Ref: Calificación proyecto de título

De mi consideración:

Habiendo revisado el trabajo de titulación "**Implementación de un Repositorio Digital empleando tecnologías de código abierto**", presentado por el alumno Sr. **GERARDO ANDRÉS MARTÍNEZ ROSAS**, mi evaluación es la siguiente:

**Nota: 6,8** (seis coma ocho).

El trabajo desarrollado por el Sr. Martínez describe de manera rigurosa y detallada la implementación de un repositorio digital en dos organizaciones, presenta pautas para su instalación y configuración y plantea la evaluación de la instalación del repositorio en las organizaciones, lo que sin duda es un aporte relevante en la línea de herramientas que potencian el trabajo colaborativo en las organizaciones. Adicionalmente, cabe destacar el aporte del trabajo en términos de la posibilidad de interconexión entre el sistema y el gestor de base de datos de la organización donde se instalará, lo que fue propuesto y probado en el desarrollo del trabajo. Entre los aspectos a mejorar, es preciso realizar algunos ajustes formales, para que no haya detalles que afecten la calidad del trabajo realizado.

<b>Aspecto</b>	<b>Evaluación</b>
Cumplimiento de objetivos	7,0
Satisfacción de alguna necesidad	7,0
Interpretación de los datos y obtención de conclusiones	7,0
Originalidad	6,5
Aplicación de criterios de análisis y diseño	7,0
Perspectivas del trabajo	7,0
Coherencia y rigurosidad lógica	6,5
Precisión del lenguaje técnico	6,5

Sin otro particular, saluda atentamente a usted,



**Mauricio Ruiz-Tagle Molina**  
Académico del Instituto de Informática  
Universidad Austral de Chile



# Universidad Austral de Chile

## Instituto de Informática

Valdivia, 23 de marzo del 2009.

De: Luis Alberto Alvarez Gonzalez.  
Informante

A: Juan Pablo Salazar Fernández.  
Director Escuela de Ingeniería Civil en Informática

Ref: Calificación proyecto de título

De mi consideración:

Habiendo revisado el trabajo de titulación "**IMPLEMENTACIÓN DE UN REPOSITORIO DIGITAL EMPLEANDO TECNOLOGÍAS DE CÓDIGO ABIERTO**", presentado por el estudiante **SR. GERARDO ANDRÉS MARTÍNEZ ROSAS**, mi evaluación es la siguiente:

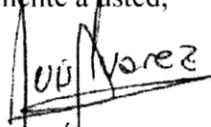
**Nota: 5,7 (CINCO COMA SIETE).**

#### Fundamento de la nota:

Su Tesis de Titulación cumple con los objetivos específicos, sin embargo, el último no guarda relación con los anteriores y perfectamente pudo ser parte del primero. Falta mayor rigurosidad en el uso de tipos de letras, referencias a figuras entre otros.

Aspecto	Evaluación
Cumplimiento de objetivos	7
Satisfacción de alguna necesidad	5
Aplicación del método científico	5
Interpretación de los datos y obtención de conclusiones	5
Originalidad	5
Aplicación de criterios de análisis y diseño	6
Perspectivas del trabajo	6,5
Coherencia y rigurosidad lógica	6
Precisión del lenguaje técnico	6

Sin otro particular, saluda atentamente a usted,

  
Luis A. Álvarez González  
Instituto de Informática

## **Dedicatoria y/o Agradecimientos**

*Este trabajo de titulación esta dedicado especialmente a mi familia, la cual ha sido un apoyo fundamental en este proceso, a mi padre Hernán Martínez siempre preocupado y apoyándome, mi gran hermano Gabriel Martínez ya que sin él no lo hubiera logrado, ya que es el apoyo fundamental en mi vida. Pero el agradecimiento mas grande es para mi madre que siempre ha estado conmigo, en cada momento que lo he necesitado, entregándome consejos, cariños y amor. Además un agradecimiento a 2 nuevos integrantes de mi familia, mi pareja y amor de mi vida Anyela Castro que me apoyó incondicionalmente, entregándome todo su amor y comprensión, además de darme el regalo mas lindo de mi vida, que es mi hijito precioso Ignacio Martínez.*

*A mis tíos(as) y primos(as) que siempre me han entregado su cariño y apoyo incondicional, además quiero agradecer especialmente a mi tía Laly, la cual, siempre ha creído en mí, desde el comienzo de este arduo camino que es llegar a ser profesional.*

*A mi mamita Carmen que siempre se ha preocupado por mí, además de criarme en los primeros años de mi vida, entregándome todo su amor y cariño.*

*A mis grandes amigos: Altidoro, Gato, Fernando, Mauro, Nano, Claudio, Alexis, Micho y Felipe que han estado incondicionalmente conmigo todos estos años, además de acompañarme en las celebraciones o a pasar las penas de fin de semestre.*

*A Carla Coronado y a su familia que me apoyaron en momentos difíciles en el comienzo de este camino.*

*A mis amigas y compañeras Tatiana, Carola y Paola, ya que junto a ellas pasamos muchas noches sin dormir con una meta en común, llegar a ser ingeniero civil en Informática. Además de Rafa que junto a él pasamos todos los ramos de ciencias básicas.*

*Gracias de todo corazón a todas las personas que estuvieron conmigo en algún momento en este proceso, ya que gracias a su apoyo he podido cumplir uno de mis grandes sueños, el ser Ingeniero Civil en Informática.*

## ÍNDICE.

ÍNDICE.....	I
ÍNDICE DE TABLAS .....	III
RESUMEN DEL PROYECTO.....	V
ABSTRACT.....	VI
1. INTRODUCCIÓN .....	1
1.1 Repositorio Digital.....	1
1.2 Fedora/Fez.....	2
1.2.1 Modelo de objetos digitales en FEDORA.....	3
1.2.2 Interacción entre el usuario y Fedora.....	4
1.3 Objetivos Generales y Específicos.....	5
1.3.1 Objetivos Generales.....	5
1.3.2 Objetivos Específicos.....	5
1.4 Convenciones de Escrituras .....	5
1.5 Estructura del Documento.....	6
2. REPOSITORIO DIGITAL .....	7
2.1 Introducción .....	7
2.2 Clasificación de Repositorios Digitales .....	8
2.2.1 Repositorio Digital Institucional.....	8
2.2.2 Repositorio Digital Temático.....	8
2.3 Estándares de metadatos .....	9
2.3.1 MARC 21 .....	9
2.3.2 Dublin Core o DC .....	10
2.3.3 MODS (Esquema de descripción de metadatos).....	12
2.3.4 RDF (Marco de Descripción de Recursos) .....	14
2.3.5 LOM.....	16
2.4 Estándar de Interoperabilidad .....	18
2.4.1 Z39.50 .....	18
2.4.2 OAI-PMH.....	19
2.5 Herramientas de Software Libre para la Implementación de Repositorios Digitales.....	21
2.5.1 CDSware .....	22
2.5.2 Dspace .....	22
2.5.3 E-Prints.....	23
2.5.4 Fedora.....	23
3. FEDORA/FEZ .....	25
3.1 Fedora ( <i>Back-end</i> ).....	25
3.1.1 Modelo Digital de Fedora .....	25
3.1.2 Arquitectura de los Servicios Fedora.....	28
3.1.3 Archivo y Preservación de los Objetos .....	30
3.1.4 Autenticación/Organización.....	30
3.2 Fez ( <i>Front-end</i> ).....	30
3.2.1 Modelo de Contenido.....	30
3.2.2 Seguridad – FezACML .....	31
3.2.3 Workflow .....	32
3.2.4 Índice de Búsqueda.....	34
3.2.5 Índices de Seguridad.....	36
3.2.6 Control de Vocabulario.....	37
3.3 Arquitectura del Repositorio Digital Fedora/Fez.....	37

4.	REPOSITORIO DIGITAL EN UNA ORGANIZACIÓN. UACH, FVT.....	39
4.1	Propuesta UACH .....	39
4.1.1	Metodología de Implementación .....	39
4.1.2	Trabajo Realizado. ....	41
4.2	Propuesta FVT. ....	42
4.2.1	Metodología de Implementación. ....	42
4.2.2	Trabajo Realizado. ....	42
5.	IMPLEMENTACIÓN DEL REPOSITORIO DIGITAL FEDORA/FEZ .....	43
5.1	Guía de Instalación de Repositorio Digital Fedora 2.1.1 y su Front-end Fez 1.3, en Sistema Operativo Fedora Core 7 .....	43
5.1.1	Requisitos.....	43
5.1.2	Actualización de Paquetes en Fedora 7.....	44
5.1.3	Instalación del Servidor Apache .....	44
5.1.4	Instalación de PHP 5 .....	45
5.1.5	Instalación de la Base de Datos MySQL.....	45
5.1.6	Instalación de phpMyAdmin.....	46
5.1.7	Instalación de J2SDK.....	47
5.1.8	Instalación de Libpng.....	47
5.1.9	Instalación de Libjpeg .....	47
5.1.10	Instalación de la Librería GD.....	48
5.1.11	Instalación de la Librería CURL .....	48
5.1.12	Instalación de la Librería Tidy .....	48
5.1.13	Instalación de ImageMagick .....	49
5.1.14	Instalación de Graphviz.....	49
5.1.15	Instalación de JHOVE.....	49
5.1.16	Instalación de domxml .....	50
5.1.17	Instalación de FEDORA-2.1.1 .....	50
5.1.18	Instalación del front-end Fez 1.3.....	53
5.2	Creación del Objeto Digital Utilizando el Estándar de Metadatos MODS.....	55
5.2.1	Editar MODS .....	56
5.2.2	Editar RELS-EXT .....	60
5.2.3	Editar FezMD.....	62
5.2.4	Editar Fedora Object XML .....	64
5.3	Creación de Índices de Búsquedas.....	66
5.4	Creación de Comunidades y Colecciones.....	67
5.5	Administración de Políticas de Seguridad para las Comunidades y Colecciones. ....	69
5.6	Creación de una API para la Conexión entre el Repositorio Digital y la Base de Datos de una Organización para la Autenticación de los Usuarios. ....	70
6.	EVALUACION DEL IMPACTO DE LA IMPLEMENTACION DEL REPOSITORIO DIGITAL EN UNA ORGANIZACIÓN.....	73
6.1	Comunidades Creadas en la Organización.....	73
6.2	Colecciones Creadas en la Organización:.....	74
6.3	Políticas de Seguridad en la Organización.....	74
6.4	Cuestionario de Evaluación.....	75
7.	PAUTAS PARA LA IMPLEMENTACION DE UN REPOSITORIO DIGITAL..	77
8.	CONCLUSIONES. ....	78
9.	REFERENCIAS BIBLIOGRAFICAS.....	80

## ÍNDICE DE TABLAS

Tablas	Páginas
Tabla 1: Sentencia Sujeto-Predicado-Objeto. ....	15
Tabla 2: Estructura del Metadato. ....	16
Tabla 3: Software necesarios para el Repositorio Digital.....	43

## ÍNDICE DE FIGURAS

Figura 1: Arquitectura de un objeto digital en FEDORA [FDT05a].....	3
Figura 2: Interacción entre Cliente y Servicio Web. [FDT05a] .....	4
Figura 3: Elemento y Atributo [FUE07] .....	13
Figura 4: Elemento y sub-elemento. [FUE07] .....	14
Figura 5: Diagrama de nodo.....	15
Figura 6: Sentencia estructurada. ....	15
Figura 7: Representación esquemática de los elementos del estándar LOM. ....	17
Figura 8: Implementación del Protocolo OAI-PMH [ECH04] .....	19
Figura 9: Modelo de Estructura del Protocolo OAI-PMH [OAFnd] .....	21
Figura 10: Estructura del Objeto Digital de Fedora. ....	27
Figura 11: Esquema del objeto digital de Fedora (FOXML). ....	27
Figura 12: Arquitectura de Fedora [PAY05] .....	28
Figura 13: Arquitectura Principal de Servicios de Fedora [PAY05] .....	29
Figura 14: Arquitectura de Servicios de Fedora [PAY05].....	29
Figura 15: Diagrama de flujo de los componentes del workflow de Fez.....	33
Figura 16: Diagrama de estados del <i>workflow</i> para la creación de registros. ....	34
Figura 17: Ingreso de una índice de búsqueda. ....	35
Figura 18: Índice de búsqueda Ingresado.....	35
Figura 19: Indexación del campo “Titulo” al índice de búsqueda “Titulo” .....	36
Figura 20: Esquema de Modelo-Vista-Controlador. ....	38
Figura 21: Diagrama de la arquitectura del repositorio Digital Fedora/Fez .....	38
Figura 22: Acceso a <i>Manage Document Type XSDs</i> .....	56
Figura 23: Despliegue de los objetos XSD MODS.....	56
Figura 24: Copia de objeto <i>Thesis</i> MODS. ....	57
Figura 25: Editando el objeto <i>Thesis</i> estándar Uach MODS. ....	57
Figura 26: Relación del sub-elementos Autor.....	60
Figura 27: Despliegue de los objetos XSD RELS-EXT. ....	60
Figura 28: Copia de Objeto Fedora Record RELS-EXT Display. ....	61
Figura 29: Editando el objeto Fedora Record Uach RELS-EXT Display. ....	61
Figura 30: Deshabilitar el elemento xml <i>resource</i> . ....	61
Figura 31: Despliegue de los objetos XSD FezMD. ....	62
Figura 32: Copia de objeto Simple FezMD Display.....	62
Figura 33: Editando el objeto Simple FezMD Uach Display. ....	63
Figura 34: Deshabilitar el elemento xml <i>refereed</i> y <i>referente_text</i> . ....	63
Figura 35: Despliegue de los objetos XSD Fedora <i>Object XML</i> .....	64
Figura 36: Copia de Objeto Generic Document.....	64
Figura 37: Editando el objeto Tesis Uach. ....	65
Figura 38: Editando el elemento xml <i>objectProperties -&gt; property-&gt;Value</i> .....	65
Figura 39: Editando el elemento xml <i>datastream-&gt;datastreamVersion-&gt;xmlContent</i> ...	66
Figura 40: Ingresar al Administrador-> <i>Manage Search Keys</i> . ....	66
Figura 41: Creación de un Índice de Búsqueda. ....	67
Figura 42: Ingresar a Lista de Comunidades-> Crear Nueva Comunidad.....	68
Figura 43: Creación de una Nueva Comunidad.....	68
Figura 44: Creación de una Nueva Colección.....	69
Figura 45: Diagrama de Autenticación de un usuario al Repositorio Digital.....	71
Figura 46: Diagrama de interacción del sistema con su propia base de datos al ingresar un nuevo usuarios al sistema.....	72
Figura 47: Diagrama de interacción del sistema con la base de datos de la organización al ingresar un nuevo usuario al sistema.....	72
Figura 48: Comunidades creadas para gestionar los documentos de FVT Ltda.....	73

## RESUMEN DEL PROYECTO

Este proyecto de tesis tiene como objetivo implementar un repositorio digital en una institución u organización, utilizando como componentes principales un gestor documental FEDORA<sup>1</sup> (*back-end*) y una Interfaz Web (*front-end*).

En el desarrollo de este proyecto de tesis se investigo los distintos sistemas de gestión documental, para así tener un mayor conocimiento de los sistemas de gestión existentes. Además se documento detalladamente la instalación de los distintos softwares necesarios para el correcto funcionamiento del sistema, ya que actualmente no existe una completa documentación para esto.

Una vez instalado correctamente el sistema se adaptaron módulos en la Interfaz Web para satisfacer los requisitos de la organización, donde se desarrollará una API para que los usuarios del sistema se autentifiquen vía remota con el servidor de la organización, y no con la base de datos del repositorio digital, basados en el marco de cooperación al proyecto Fedora. Además se generaron pautas que permitirán la implementación de repositorios digitales en las organizaciones, y se evaluó el impacto que causa la implementación de un repositorio digital sobre estas. Para así obtener ventajas competitivas a nivel de capital intelectual.

---

<sup>1</sup> Es un sistema de repositorio digital desarrollado conjuntamente por la Universidad de Cornell y la Universidad de Virginia, FEDORA son las siglas de *Flexible Extensible Digital Object Repository Architecture* (Arquitectura Repositorio Objeto Digital Flexible Extensible)

## **ABSTRACT**

This thesis project aims to implement a digital repository on an institution or organization, using as a main component manager documentary FEDORA (back-end) and a Web interface (front-end). In developing this thesis project investigate the various document management systems, this way to have a major knowledge of the existing systems of management. In addition there receive documents detailed the installation of the different softwares necessary for the correct functioning of the system, as currently there is no complete documentation for this.

Once installed correctly the system be adapted modules in the web interface to satisfy the requirements of the organization, where develop an API for that the users of the system will be authenticate with the server via remote from the organization, and not with the data base of the digital repository, based on the frame of cooperation to the project FEDORA. In addition generate guidelines that will allow the implementation of digital repositories in organizations, and there be evaluated the impact that causes the implementation of a digital repository of these. To get a competitive advantages at the level of intellectual capital.

# 1. INTRODUCCIÓN.

## 1.1 Repositorio Digital.

En la actualidad, en las organizaciones es cada vez más importante poder gestionar de forma eficaz y eficiente, los datos relacionados a sus procesos productivos y/o capital intelectual. Para así, obtener ventajas competitivas y alcanzar un avanzado estado de madurez organizacional. Para alcanzar tales objetivos existen los sistemas llamados repositorios digitales.

Un repositorio digital es un sistema completo para la preservación en el tiempo del contenido digital y/o documentación generada por una organización. Este sistema incluye la plataforma, el sistema de almacenamiento, el gestor documental y las políticas institucionales referidas al repositorio.

Las principales características de un repositorio digital son:

- Contiene objetos digitales
- Contiene metadatos<sup>2</sup>.
- Asegura la identificación persistente del objeto mediante un identificador único persistente.
- Ofrece funciones de gestión, archivo y preservación de los objetos.
- Proporciona un acceso fácil, controlado y estandarizado a los objetos.
- Ofrece los sistemas adecuados de seguridad para los objetos y los metadatos.
- Sostenible en el tiempo.

Un gestor documental es un sistema para la organización y administración del contenido de un repositorio digital.

Existen varios sistemas de gestión documental en la actualidad, tales como, DSpace [DSP], Fedora [FED], EPrints [EPR], i-Tor [ITO] y CDSware. [CDS]

---

<sup>2</sup> Los Metadatos son datos altamente estructurados que describen información acerca de un recurso digital.

## 1.2 Fedora/Fez

Entre los sistemas de gestión documental más utilizados (DSpace, Fedora, EPrints, i-Tor, y CDSware), el que satisface en mayor grado las características de escalabilidad, seguridad, interoperabilidad, facilidad de despliegue, código libre y herramienta *workflow* es el sistema de gestión documental FEDORA, esto se verá con mayor detalle el capítulo 2 de este documento.

El sistema de gestión documental **FEDORA** es un sistema para repositorios digitales basados en estándares abiertos (XML<sup>3</sup>, PHP<sup>4</sup>, MySQL<sup>5</sup>). Puede ser usado como arquitectura base, para cualquier repositorio institucional y biblioteca digital basada en tecnología Web. [FDT05a]

Para la interacción entre el sistema de gestión documental Fedora (*back-end*) con el usuario, existe un software llamado **Fez**<sup>6</sup>, el cual, provee un completo, amigable, flexible y altamente configurable manejo de un repositorio digital y *workflow*, basado en la plataforma FEDORA 2, orientado a distintos tipos de contenido, comunidades y colecciones. Actualmente es desarrollado por la Universidad de Queensland, Australia, como parte de la iniciativa gubernamental del ministerio de educación australiano llamado APSR<sup>7</sup>. [KOR06]

El sistema de gestión documental FEDORA posee las siguientes características:

- Sólido modelo de objetos digitales.
- Gestión de Metadatos extensible.
- Integración de servicios Web.
- Gestión de versiones.
- Arquitectura de seguridad

---

<sup>3</sup> XML siglas en inglés de *Extensible Markup Language* (Lenguaje de Marcas Extensibles).

<sup>4</sup> PHP es un lenguaje de programación, es un acrónimo recursivo que significa PHP *Hypertext Pre-processo*.

<sup>5</sup> MySQL es un sistema de gestión de base de datos relacional, multihilo y multiusuario.

<sup>6</sup> Fez es una interfaz Web que Administra el sistema de repositorio digital Fedora.

<sup>7</sup> APSR siglas en inglés de *Austalian Partnership for Sustainable Repositories*.

### 1.2.1 Modelo de objetos digitales en FEDORA

Un objeto de datos en FEDORA describe el contenido, y un *set* de comportamientos o servicios asociados, que pueden ser aplicados al contenido.

La figura 1 muestra la arquitectura de un objeto digital en Fedora.

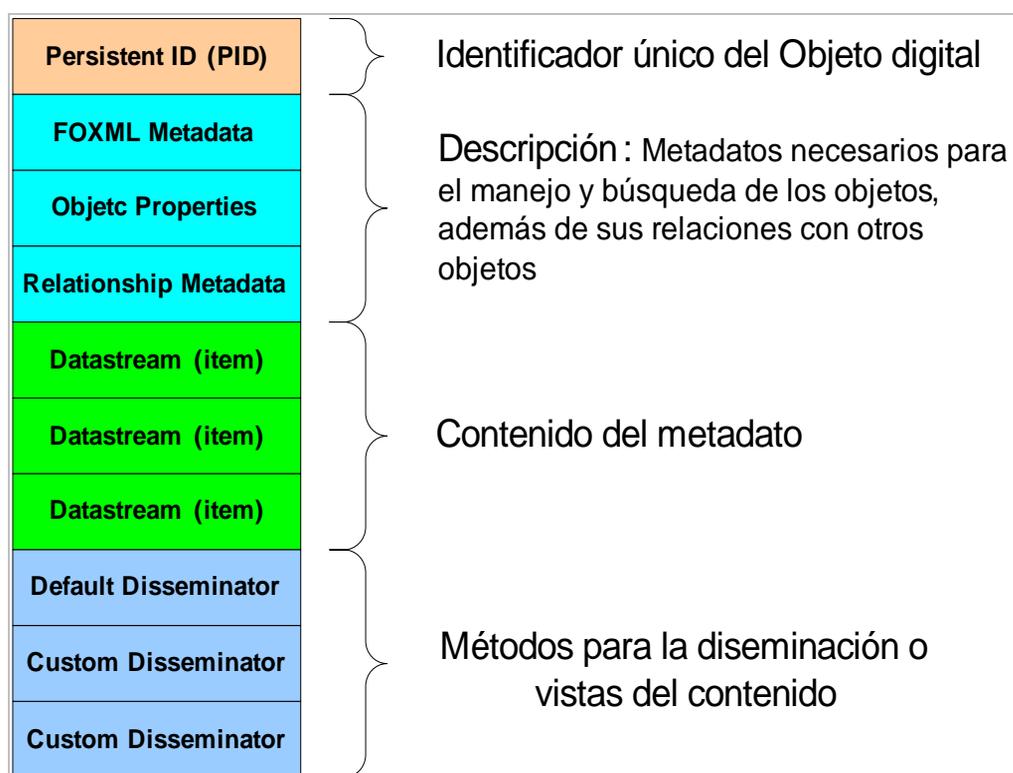


Figura 1: Arquitectura de un objeto digital en FEDORA [FDT05a]

Los objetos digitales en Fedora consisten en 4 partes:

1. **Identificador del Objeto Digital:** Es un Identificador único, persistente para el objeto digital (Persistent ID).
2. **Perspectiva Descriptiva:** FOXML<sup>8</sup> es metadato de un objeto digital, es el metadato que debe ser registrado con cada objeto digital, para facilitar el manejo de estos objetos. El metadato FOXML es distinto a otros metadatos que son almacenados en el objeto digital como por ejemplo el contenido. Este tipo de metadato es el requerido por la arquitectura de Fedora. Todos los otros metadatos (Ej: metadato descriptivo, metadato técnico) son considerados opcionales del punto de vista del repositorio, y son tratados como un *datastream* en un objeto digital. Las propiedades del Objeto (*Object Properties*) describen el tipo del objeto, su estado, fecha de creación, últimas modificaciones del objeto y su etiqueta.

<sup>8</sup> FOXML siglas en inglés de Fedora Object XML (Objeto de Fedora en XML).

La relación de metadato (*Relationship Metadata*) describe cualquier relación que existe entre objetos digitales en un repositorio Fedora.

3. **Item de metadatos:** un *datastream* es el componente de un objeto digital que representa el contenido digital. En otras palabras, un *datastream* representa el material digital que es la esencia de un objeto digital (Ej: imágenes digitales, texto codificado, grabación de audio).
4. **Perspectiva de servicio:** Un “*disseminator*” es un componente del objeto digital que es usado para asociar comportamientos (Ej: Servicios) al objeto digital. Un “*disseminator*” apunta a un conjunto de métodos que son invocados para producir representaciones virtuales del objeto. [FDT05a]

### 1.2.2 Interacción entre el usuario y Fedora.

Como se muestra en la figura 2, los usuarios sólo interactúan con el repositorio digital mediante las APIs<sup>9</sup>, aún cuando a veces pueda parecer que ellos actúan directamente con un objeto, esto no es así.

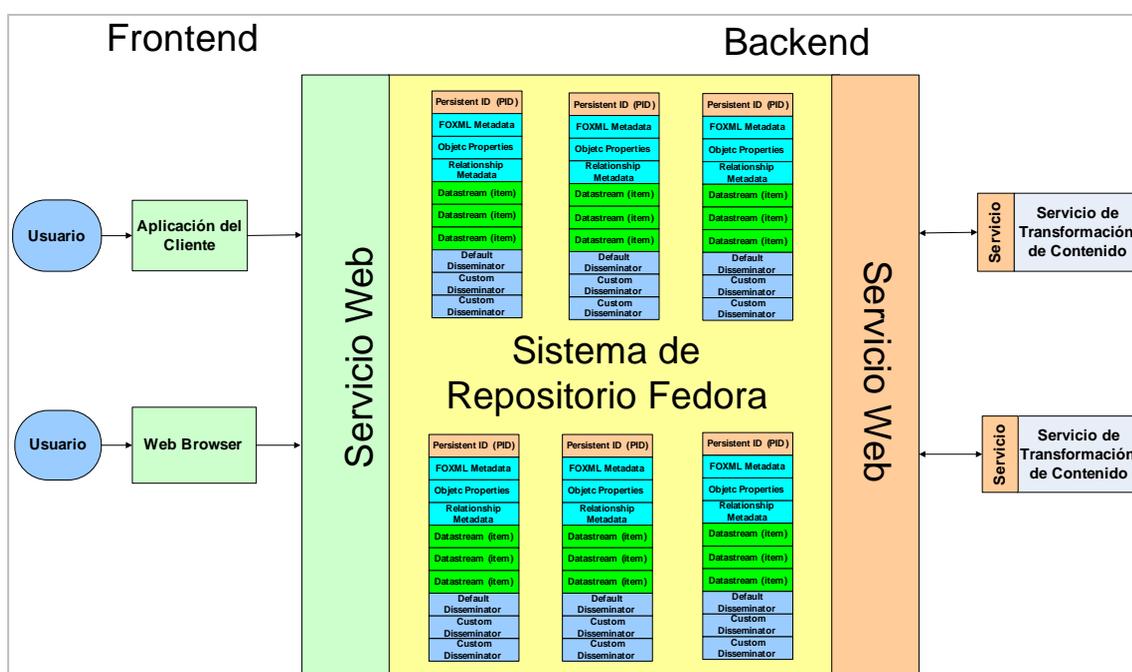


Figura 2: Interacción entre Cliente y Servicio Web. [FDT05a]

<sup>9</sup> API siglas en inglés de *Application Programming Interface* (Interfaz de Programación de Aplicaciones).

## **1.3 Objetivos Generales y Específicos**

### **1.3.1 Objetivos Generales.**

Implementar un repositorio digital, utilizando como componentes principales el gestor documental FEDORA (*back-end*) y una Interfaz Web (*front-end*), con el objeto de cooperar en el perfeccionamiento del proyecto Fedora y generar pautas que permitan la implementación de un repositorio digital en una organización.

### **1.3.2 Objetivos Específicos.**

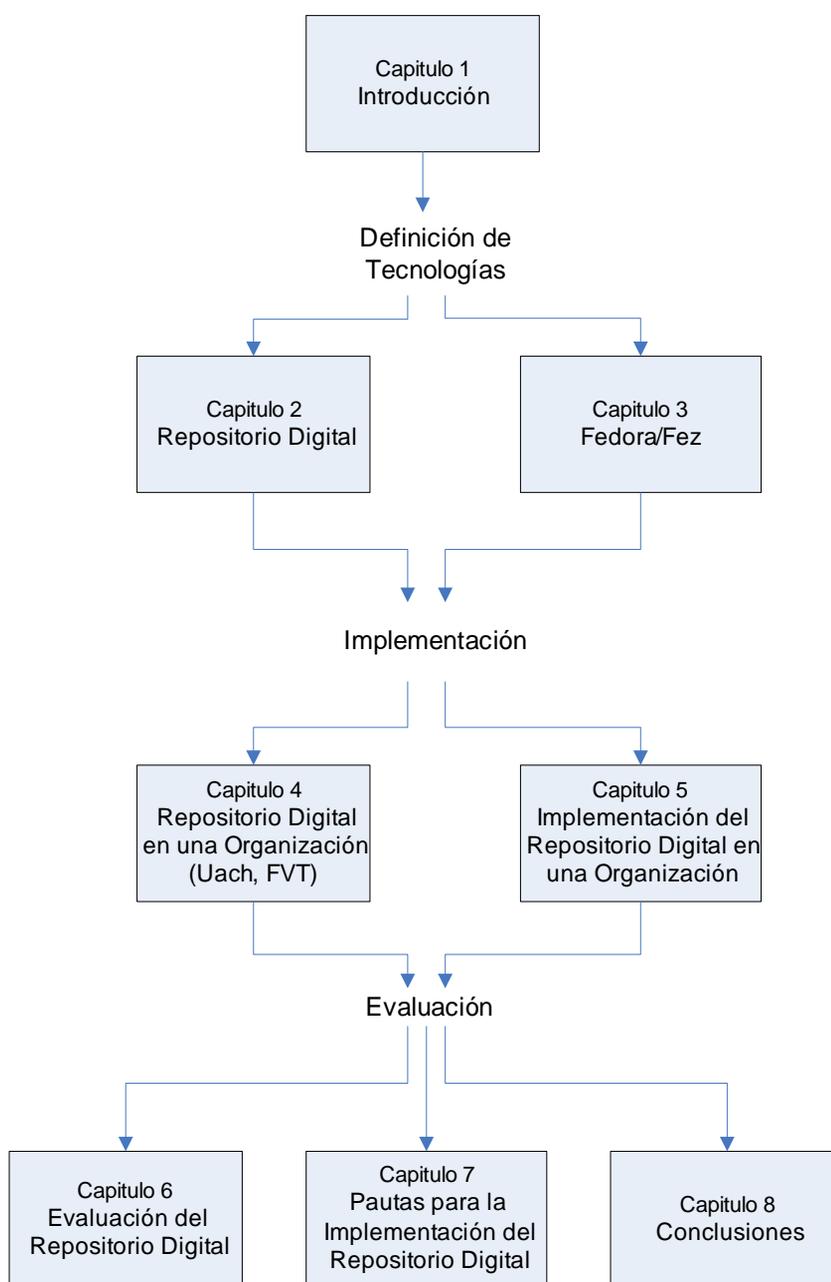
1. Describir tecnologías que permitan la implementación de repositorios digitales en una organización.
2. Seleccionar una organización e implementar un repositorio digital utilizando como gestor documental Fedora, junto con el desarrollo de pautas para la instalación y un módulo o plugin que permita la interconexión con sistemas externos vía XML para la Interfaz Web Fez, basados en el marco de cooperación del Proyecto Fez/Fedora.
3. Proponer pautas que permitan la implementación de repositorios digitales en una organización.
4. Evaluar el impacto de la implementación del repositorio digital en la organización.

## **1.4 Convenciones de Escrituras**

Las referencias son escritas de la forma [ABC01], donde las tres letras representan el apellido del autor del libro, artículo, institución o texto usado como referencia y los números el año de publicación, si no se obtiene la fecha de publicación la referencia será de la forma [ABC]. En el caso de que un documento haya sido hecho por más de un autor, se considerará el apellido del primer autor, en forma alfabética, en caso de no definirse un autor principal de la obra referida. En los casos en que exista más de una publicación del mismo autor y el mismo año, se utilizará una letra minúscula, de forma correlativa, a continuación del año.

Las imágenes y diagramas serán referenciadas como Figura n, siendo n un número correlativo único asociado a cada imagen o diagrama. Términos en otros idiomas serán denotados por letras cursivas.

## 1.5 Estructura del Documento



## 2. REPOSITORIO DIGITAL

### 2.1 Introducción

En la actualidad para las empresas u organizaciones la información y los contenidos digitales cada vez son más importantes, donde los datos relacionados a sus procesos productivos y capital intelectual son muy importantes para obtener ventajas competitivas. Por ejemplo, para las universidades los contenidos digitales son fundamentales para los procesos de investigación académica, la docencia y la administración.

Las organizaciones generan grandes cantidades de información, donde los contenidos digitales tienen diversos formatos, tales como: documentos de texto (publicaciones electrónicas de libros, revistas, tesis, conferencias), audio, imágenes, videos, mapas, etc. Para facilitar el acceso, la gestión, almacenamiento, búsqueda y recuperación de los contenidos digitales, evitando los cuellos de botella, acumulación de información en puntos susceptibles a fallas, problemas de seguridad y conflictos por acceso de personal no autorizados, se han creado los repositorios digitales.

Un repositorio digital es un sistema completo para la preservación en el tiempo del contenido digital y/o documentación generada por una organización. Este sistema incluye la plataforma, el sistema de almacenamiento, el gestor documental y las políticas institucionales referidas al repositorio.

Las principales características de un repositorio digital son:

- Contiene objetos digitales
- Contiene metadatos.
- Asegura la identificación persistente del objeto mediante un identificador único persistente.
- Ofrece funciones de gestión, archivo y preservación de los objetos.
- Proporciona un acceso fácil, controlado y estandarizado a los objetos.
- Ofrece los sistemas adecuados de seguridad para los objetos y los metadatos.
- Sostenible en el tiempo. [LOP07]

**Objeto Digital:** Un objeto digital es un registro de información codificado digitalmente, el cuál contiene información (tesis, revistas, imagen, audio y libros), metadatos, un identificador y una serie de comportamientos sobre su contenido.

**Metadatos:** Los metadatos son datos altamente estructurados que describen información acerca de un recurso digital, es esencialmente “información acerca de la información” o “datos sobre datos”, facilitando la descripción y recuperación del mismo. Los metadatos están estructurados por un número de elementos, tales como: título, autor, fecha de creación, etc. Dichos elementos que conforman los metadatos, están definidos por algún estándar, así cuando los usuarios que deseen compartir metadatos, estarán de acuerdo con el significado preciso de cada elemento. [INE03]

Algunos beneficios de los metadatos son:

- Describen la información existente.
- Organizan el conocimiento.
- Facilitan la búsqueda y la utilización de dicha información y/o conocimiento.
- Proveen información necesaria para interpretar y procesar datos transferidos por otra organización.

**Identificador único:** Es un identificador único que identifica al objeto digital.

## **2.2 Clasificación de Repositorios Digitales**

Existen 2 tipos de repositorios digitales, los cuales son:

### **2.2.1 Repositorio Digital Institucional**

Un repositorio digital institucional es aquel creado, mantenido y autorizado por una institución (no exclusivamente una universidad) o un grupo de instituciones, que recoge los contenidos digitales generados por la actividad de los miembros de esa institución.

Un repositorio institucional es un conjunto de servicios que una institución ofrece a su comunidad para la gestión, y difusión de los contenidos digitales generados por los miembros de esa comunidad. Es, en su nivel más básico, un compromiso organizativo para el control de esos materiales digitales, incluyendo su preservación, su organización, acceso y distribución. [Cli03]

### **2.2.2 Repositorio Digital Temático**

Un repositorio digital temático reúne sus contenidos en función de su área temática, no por su origen institucional, en torno a una disciplina científica o disciplinas relacionadas.[LOP07]

## 2.3 Estándares de metadatos

Es importante en la gestión de recursos de información, los esquemas y sistemas de etiquetado de la información para una mayor comunicación entre distintos sistemas.

En la actualidad existen varios estándares para los metadatos, pero los más importantes y utilizados para la implementación de repositorios digitales son:

- MARC 21<sup>10</sup>
- Dublin Core
- MODS<sup>11</sup>
- RDF<sup>12</sup>
- LOM<sup>13</sup>

Estos estándares soportan su formulación dentro de XML (excepto MARC 21), además todos ellos están pensados para ofrecer soporte a la descripción de documentos digitales de cualquier tipo.

### 2.3.1 MARC 21

Es un estándar basado en una estructura de formato ISO 2709. Este estándar permite codificar y transmitir un gran número de datos de una forma estandarizada, fácilmente procesable y una expresividad o nivel de detalle necesario para la descripción de recursos.

MARC 21 es un estándar extensamente usado para la representación y cambio de datos de Autoridad, datos Bibliográficos, datos de Clasificación, Información de Comunidad, y datos de Propiedad. Por lo tanto MARC 21 es una familia de 5 formatos coordinados, los cuales son:

1. MARC 21 *Format for Authority Data*.
2. MARC 21 *Format for Bibliographic Data*.
3. MARC 21 *Format for Classification Data*.
4. MARC 21 *Format for Community Information*.
5. MARC 21 *Format for Holdings Data*.

El formato MARC 21 para datos Bibliográficos (*MARC 21 Format for Bibliographic Data*) está diseñado para contener información bibliográfica, tal como

---

<sup>10</sup> MARC 21 siglas en inglés de *Machine Readable Cataloging* (Registros catalográficos legibles por máquina).

<sup>11</sup> MODS siglas en inglés de *Metadata Object Description Schema* (Metadatos esquema de descripción de objetos).

<sup>12</sup> RDF siglas en inglés de *Resource Description Framework* (Marco de Descripción de Recursos).

<sup>13</sup> LOM sigla en inglés de *Learning Object Metadata*.

títulos, nombres, temas, notas, información sobre publicación, y descripción física de ítems.

Los registros MARC están compuestos por tres elementos:

1. **La estructura del registro:** Es una implementación de ISO2709<sup>14</sup>.
2. **La designación de contenido:** Es el conjunto de etiquetas y códigos que identifican y caracterizarán a los elementos de datos dentro de un registro, y darán soporte a la manipulación del contenido de datos.
3. **El contenido de datos de registro:** El contenido de los elementos de datos que comprenden un registro MARC generalmente están definidos por normas externas a los formatos, tales como las reglas de catalogación, los sistemas de clasificación, listas de códigos, y otras convenciones usadas por la organización que crea un registro.

La desventaja que existe con el estándar MARC 21 es que su estructura para el registro bibliográfico no está basada en XML. Esto se ha vuelto una necesidad, ya que en la actualidad muchos protocolos nuevos están más “cómodos” con una estructura para el registro bibliográfico basada en XML. En respuesta a esa necesidad, en estos últimos años han sido desarrolladas estructuras alternativas para los elementos de datos en MARC 21. La más importante es MARCXML, que provee un camino de ida y vuelta sin interrupción de MARC 21 a MARC en XML. [LOC99a]

### 2.3.2 Dublin Core o DC

Es un estándar de metadatos desarrollado por la DCMI<sup>15</sup>, que consta de un conjunto de 15 definiciones semánticas descriptivas que permiten la descripción de cualquier recurso de información digital, describiendo aspectos como, derechos de propiedad, autoría y contenido.

Los 15 elementos que contiene el estándar Dublin Core se pueden clasificar en 3 grupos, que indican la clase o el ámbito de la información que se almacenarán en ellos:

- **Elementos relacionados principalmente con el contenido del recurso:**

1. **Title** (Título): Es el nombre dado a un recurso.

---

<sup>14</sup> ISO2709 es un estándar de ISO para las descripciones bibliográficas.

<sup>15</sup> DCMI siglas en inglés de *Dublin Core Metadata Initiative*.

2. **Subject** (Tema): Es el tema del contenido del recurso. Un tema será expresado como una palabra clave o frases claves que describirán el tema de un recurso.
  3. **Description** (Descripción): Es una descripción del contenido del recurso. Puede ser un resumen en el caso de un documento, o una descripción del contenido en el caso de un documento visual.
  4. **Source** (Fuente): Es una referencia a un recurso del cuál se deriva el recurso actual.
  5. **Languaje** (Idioma): Define el idioma del contenido intelectual del recurso, Para ésto, se debe usar RFC 3066<sup>16</sup> en conjunto con la ISO 639<sup>17</sup>, que definen las etiquetas de dos o tres letras primarias para lenguaje, con sub-etiquetas opcionales.
  6. **Relation** (Relación): Es un identificador de la relación entre un segundo recurso con el actual. Este elemento permite enlazar los recursos con sus descripciones relacionadas.
  7. **Converage** (Cobertura): Esta una cobertura de Localización Espacial (un nombre de lugar o coordenada geográfica) y Período Temporal (fecha o rango de fecha) o Jurisdicción (nombre de una entidad) del contenido intelectual del recurso. El período temporal es sobre el contenido del recurso no de la fecha de creación.
- **Elementos relacionados principalmente con el recurso cuando es visto como una propiedad intelectual.**
    8. **Creator** (autor): Es la persona u organización responsable de la creación del contenido intelectual del recurso.
    9. **Publisher** (Editor): Es la entidad responsable de hacer que el recurso se encuentre disponible.
    10. **Contributor** (Otros Autores/Colaboradores): Son las personas u organizaciones que hayan colaborado al contenido del recuso.
    11. **Rights** (Derechos): Es una referencia (Ej: URL) sobre la información de los derechos de propiedad y del recurso.
  - **Elementos relacionados principalmente con las instancias del recurso:**
    12. **Date** (Fecha): Es la fecha donde el recurso se puso a disposición del usuario.

---

<sup>16</sup>RFC 3066 es el estándar más difundido para la identificación de idiomas en documentos de Internet

<sup>17</sup>ISO 639 es un estándar ISO que da códigos para representar los idiomas y grupos o familias de idiomas.

13. **Type** (Tipo del Recurso): Es la naturaleza o categoría del contenido del recurso.
14. **Format** (Formato): Es el formato de datos del recurso.
15. **Identifier** (Identificador): Es una secuencia de caracteres utilizados para identificar unívocamente un recurso. [DCM08]

### 2.3.3 MODS (Esquema de descripción de metadatos)

Es un estándar de metadatos desarrollados por la Oficina de Desarrollo de Redes y Normas MARC de la Biblioteca del Congreso.

Este estándar consiste en un esquema para codificar metadatos e información bibliográfica utilizando XML, el cuál, tiene el mismo objetivo que el estándar Dublin Core, pero MODS busca superar los problemas y limitaciones detectados en Dublin Core. Para ello, toma como base el formato MARC, del que selecciona diferentes elementos. Además MODS utiliza nombres de etiquetas en lugar de etiquetas numéricas, reagrupando elementos del formato bibliográfico MARC 21.

En su última versión MODS 3.3, contiene 20 elementos de nivel superior, donde cada uno de ellos contiene sub-elementos o elementos hijos que matizan su semántica, además contiene 2 elementos Raíz.

Los elementos de nivel superior son los siguientes:

1. **titleInfo**: Contiene al menos un elemento hijo llamado *title*.
2. **name**: Es el “encabezamiento”. Puede ser una persona, organización o conferencia. Este elemento puede incluir los elementos hijos: *namePart*, *role*, *affiliation* y *displayForm*.
3. **typeOfResource**: Es el Tipo de Recurso.
4. **genre**: Es el género del Contenido. No tiene elementos hijos.
5. **originInfo**: Es la Información acerca del origen del recurso, incluyendo lugar de origen o publicación, editor/autor, y fechas asociadas con el recurso.
6. **language**: Es el idioma en el cual esta el contenido del recurso.
7. **physicalDescription**: Es un elemento que contiene todos los sub-elementos que se relacionan con la información de descripción física del recurso descrito.
8. **abstract**: Es un resumen del contenido del recurso.
9. **tableOfContents**: Es una descripción del contenido del recurso. Incluirá una URL que apunta a un recurso con esta información.
10. **targetAudience**: Describe la audiencia a la cuál va dirigido el contenido.

11. **note:** Información general textual relacionada con el recurso. Puede apuntar a una nota disponible como un recurso externo mediante un hipervínculo.
12. **subject:** Es un término o frase que representa el tema principal en el que está enfocado el recurso.
13. **classification:** Es un código de clasificación, según un esquema de clasificación que se indicará en el atributo “*authority*” de éste elemento.
14. **relatedItem:** Es la Información que identifica otros recursos relacionados con la descripción del recurso.
15. **identifier:** Es un número estándar o código único que identifica al recurso.
16. **location:** Identifica la Institución o Repositorio que contiene al recurso.
17. **accessCondition:** Información acerca de las restricciones de acceso del recurso.
18. **part:** Es la designación de las partes físicas del recurso, en forma detallada.
19. **extension:** Es la información adicional del recurso.
20. **recordInfo:** Es la información sobre el registro de metadato.

Los Elementos Raíz son los siguientes:

1. **mods:** Es un registro MODS
  2. **modsCollection:** Es una colección de registros MODS
- [LOC99a] [FUE07]

La figura 3 muestra la sintaxis de un elemento con su atributo:

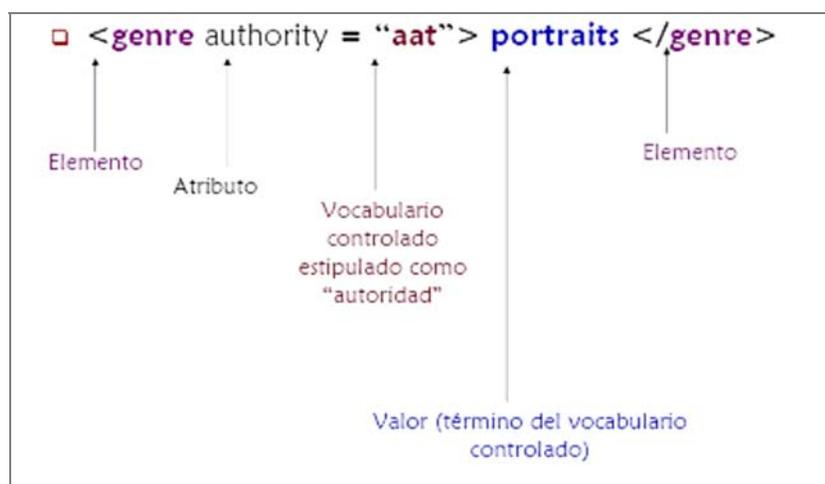


Figura 3: Elemento y Atributo [FUE07]

La figura 4 muestra la sintaxis de un elemento con su sub-elemento:

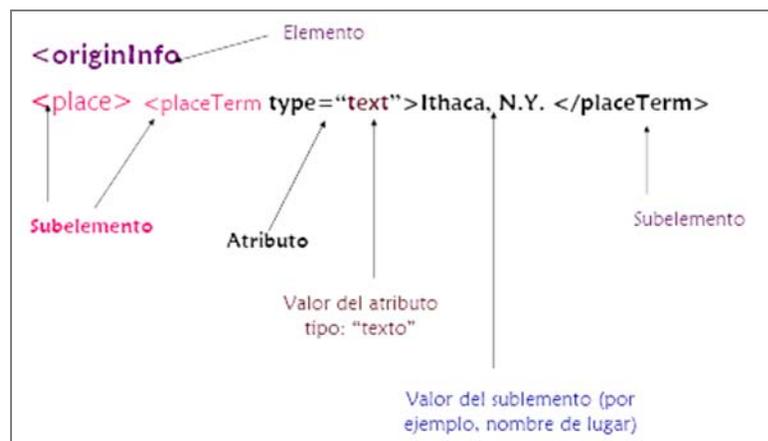


Figura 4: Elemento y sub-elemento. [FUE07]

### 2.3.4 RDF (Marco de Descripción de Recursos)

Es un fundamento para el procesamiento de metadatos, proporcionando interoperatividad entre aplicaciones que intercambian información, fue desarrollado por la W3C<sup>18</sup>.

RDF utiliza XML para intercambiar descripciones de recursos Web, donde los recursos descritos pueden ser de cualquier tipo, incluyendo recursos XML y no-XML.

El objetivo general de RDF es definir un mecanismo para describir recursos que no esté limitado sobre un dominio de aplicación particular, ni defina (a priori) la semántica de algún dominio de aplicación. La definición debe ser neutral con respecto al dominio, sin embargo, el mecanismo debe ser adecuado para describir información sobre cualquier dominio.

El modelo de datos de RDF se basa en la conversión de las declaraciones de los recursos, en sentencias de la forma Sujeto-Predicado-Objeto. Donde el Sujeto es el recurso, es decir aquello que se está describiendo. El Predicado es la propiedad o relación que se desea establecer acerca del recurso. Por último, el Objeto es el valor de la propiedad o el otro recurso con el que se establece la relación. Como las propiedades RDF representan también la relación entre recursos, los modelos RDF pueden parecer diagramas de Entidad-Relación.

Por ejemplo, una sentencia simple sería de la siguiente forma:

Juan Pérez es el creador del recurso <http://www.uach.cl/infoalumnos/siveduc>

Esta sentencia comprende las siguientes partes:

<sup>18</sup>.W3C siglas en inglés de *World Wide Web Consortium*

Tabla 1: Sentencia Sujeto-Predicado-Objeto.

Sujeto (Recurso)	http://www.uach.cl/infoalumnos/siveduc
Predicado (Propiedad)	Creator
Objeto (Valor)	Juan Pérez

Una representación gráfica de la sentencia RDF, utilizando diagrama de nodos, dónde el nodo circular representa un recurso, la flecha representa la propiedad y el nodo rectangular el valor de la sentencia, quedaría como la figura 5:

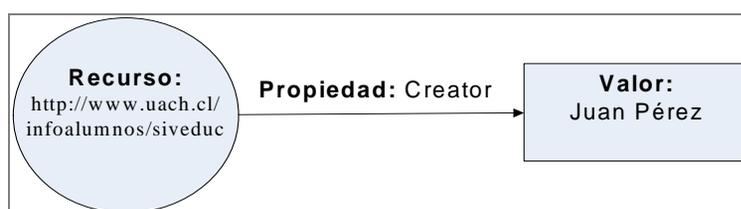


Figura 5: Diagrama de nodo.

Otro ejemplo sería una sentencia estructurada, la cuál, se muestra a continuación:

El identificador 9559 se refiere al Profesor Juan Pérez, quién tiene la dirección de correo jperez@uach.cl. Esta persona es el creador del recurso http://www.uach.cl/infoalumnos/siveduc.

El modelo para esta sentencia RDF es el de la figura 6:

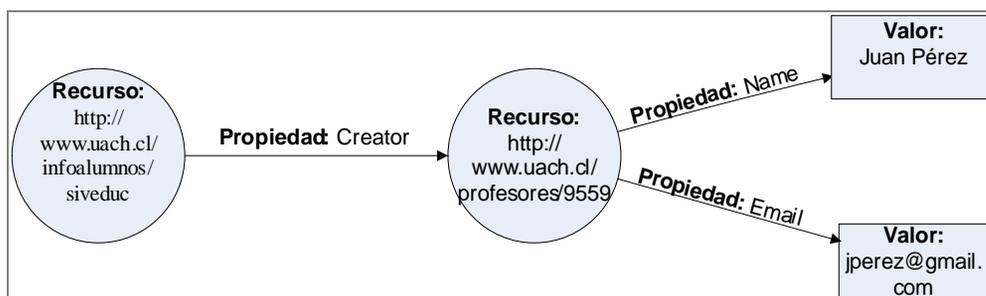


Figura 6: Sentencia estructurada.

RDF tiene distintas áreas de aplicación, como por ejemplo:

- Recuperación de datos (proporcionando mejores prestaciones a los motores de búsqueda)
- Catalogación en bibliotecas digitales (especificando también las relaciones de contenido disponibles en un sitio Web determinado).
- Agentes inteligentes (facilitando el intercambio de conocimiento).

- Sistema de Gestión de propiedad intelectual (expresando políticas de privacidad de un determinado objeto, etc.).

[W3C04a]

### 2.3.5 LOM

Es un estándar para los metadatos de objetos de aprendizaje, el cual, fue desarrollado por el Comité de Estándares para Tecnologías de Aprendizaje de IEEE 1484.12.1:2002 (LTSC), en él participaron más de 50 expertos y la propuesta tiene su origen en los proyectos ARIADNE e IMS.

Los Metadatos de Objetos de Aprendizaje (LOM), usualmente codificado en XML, son etiquetas que se le colocan a un objeto de aprendizaje para describir la información que contiene, con el uso de metadatos se pueden catalogar.

Se proponen 9 familias de etiquetas, las cuales se muestran en la Tabla 2.

Tabla 2: Estructura del Metadato.

	<b>Categoría</b>	<b>Descripción</b>
<b>1</b>	General	Descripción del Objeto (ej: descripción, identificador, título, lenguaje, etc.)
<b>2</b>	Ciclo de Vida	Aquí se describe la historia y estado actual del objeto (versión, estado,...)
<b>3</b>	Meta-Metadatos	Información acerca del propio metadato.
<b>4</b>	Técnica	Características técnicas del Objeto de Aprendizaje.
<b>5</b>	Educacional	Características pedagógicas de los Objetos de Aprendizaje
<b>6</b>	Derechos	Condiciones de uso.
<b>7</b>	Relación	Define relación entre Objetos de Aprendizaje
<b>8</b>	Anotación	Comentarios de uso educacional
<b>9</b>	Clasificación	Describe el Objeto de Aprendizaje en relación a un sistema de clasificación general.

1. **General:** Describe información general del Objeto de aprendizaje como un identificador único, nombre del Objeto de Aprendizaje, título, lenguaje en el que fue escrito, descripción, palabras claves, periodo de tiempo, cultura o región geográfica al que se refiere, estructuras del objeto y nivel de agregación o granularidad.
2. **Ciclo de Vida:** Se refiere a la historia, evolución y estado actual del Objeto de Aprendizaje.

3. **Meta-Metadatos:** Tienen los mismos tipos de etiquetas de la Categoría General, pero referido a la instancia que genero el LO.
4. **Técnicas:** Se refiere a las características técnicas del LO, como por ejemplo: formato (ej: Audio/mp3), tamaño en bytes, localización, requerimientos técnicos para su uso, tipo de tecnología requerida (hardware, software, red,..), descripción del proceso de instalación y duración.
5. **Educacional:** Se describe todas las características pedagógicas del LO, por ejemplo el tipo de interactividad, nivel de interactividad, tipo de recurso de aprendizaje, densidad semántica, rol del usuario, contexto en el cual debe ser usado, rango de edad típico, nivel de dificultad y tiempo de aprendizaje típico.
6. **Derecho:** Son referidas a la propiedad intelectual, derechos de propiedad, condiciones de uso y costo de LO.
7. **Relación:** Aquí se define la relación del LO con otros.
8. **Anotaciones:** En esta categoría se escriben los comentarios del uso educacional del LO e información referida a la fecha y hora en que se colocaron los comentarios.
9. **Clasificación:** Se describe bajo que tipo o taxonomía se clasifica el LO, se indica las referencias de la clasificación.

En la figura 7 se muestra una representación esquemática de los elementos de estándar LOM.

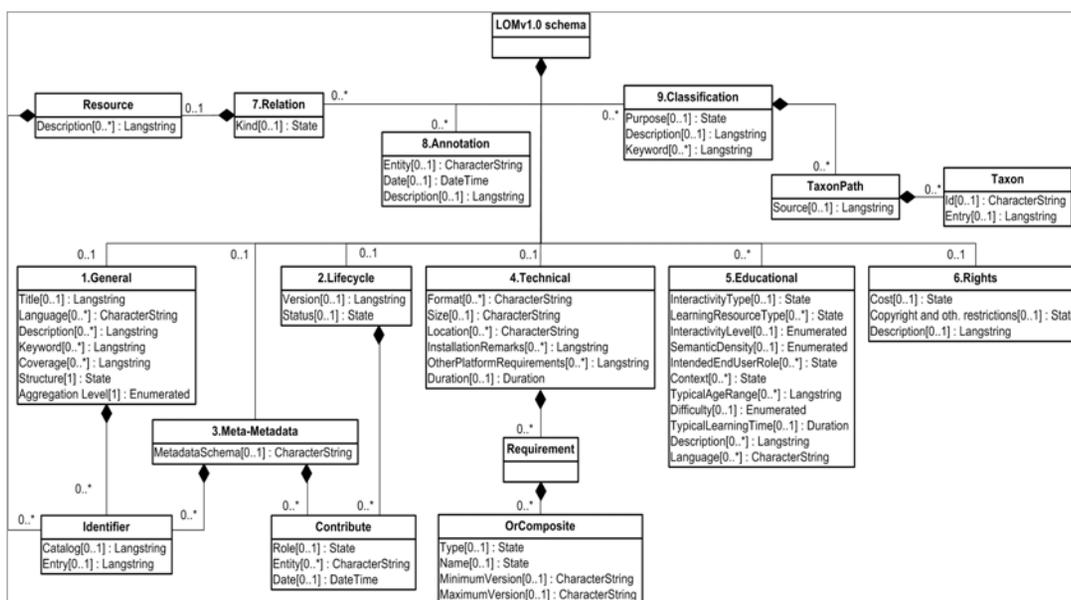


Figura 7: Representación esquemática de los elementos del estándar LOM

## 2.4 Estándar de Interoperabilidad

Los principales estándares de interoperabilidad para los repositorios digitales son:

- Z39.50

- OAI-PMH

### 2.4.1 Z39.50

El protocolo Z39.50 deriva su nombre de haber sido desarrollado por el comité número 39 de la ANSI<sup>19</sup>, y por ser el estándar número 50 de la NISO<sup>20</sup>. Este protocolo soluciona los problemas asociados a la búsqueda en múltiples bases de datos con diferentes lenguajes y procedimientos. Para ello, utiliza el modelo de búsqueda distribuida.

El Z39.50 es un protocolo para la recuperación de información basado en la estructura cliente/servidor, el cuál, facilita la interconexión entre los usuarios y las bases de datos donde se encuentra la información que necesitan. A partir de una interfaz común y de fácil manejo, independiente del lugar en que las bases de datos se encuentren, de la estructura de la base de datos, y de la forma de acceso. Además éste protocolo fue desarrollado y mantenido para bibliotecas, donde su objetivo principal es hacer posible la búsqueda y recuperación de información, en una base de datos normalmente bibliográfica.

Por lo tanto éste protocolo provee un estándar para:

- El intercambio de mensajes entre el sistema origen (servidor) y objetivo (cliente).
- La estructura y la semántica de la ecuación de búsqueda.
- La secuencia del intercambio de mensajes.
- El mecanismo para devolver los resultados.

La Tecnología Z39.50 es simple, aunque en algunos casos las aplicaciones son complejas. Hay 3 puntos importantes de las aplicaciones con este protocolo que han cambiado la actividad bibliotecaria:

1. Los modelos clientes Z39.50 pueden enviar solicitudes a varias bibliotecas simultáneamente, ya sea dentro de una misma solicitud o en varias independientes, permitiendo un ahorro de tiempo.
2. El formato básico usado para el intercambio de registros bibliográficos es el MARC. La capacidad de presentar y transferir en formato MARC, permite al cliente utilizar esa información para un procesamiento posterior.

---

<sup>19</sup>ANSI siglas en inglés de *American National Standards Institute* (Instituto Nacional Americano de Estándares).

<sup>20</sup>NISO siglas en inglés de *National Information Standards Organization* (Organización Nacional de Información de Estándares).

3. A través del uso de Z39.50 muchos otros procesos bibliotecarios han llegado a ser abiertos, particularmente, la conmutación bibliográfica y el préstamo bibliotecario. [BTR01] [BENnd]

### 2.4.2 OAI-PMH<sup>21</sup>

Es un protocolo utilizado para la transmisión de metadatos en Internet, el cuál utiliza el modelo de recolección de metadatos (*metadata harvesting*) en vez del modelo de búsqueda distribuida (ej: Z39.50).

El modelo de recolección de metadatos permite a los proveedores de documentos electrónicos exponer sus metadatos a través de una interfaz, con el objetivo de que la misma pueda ser utilizada como base para desarrollar servicios de valor añadido.

Básicamente OAI-PMH utiliza transacciones http<sup>22</sup> para emitir preguntas y obtener respuestas entre un servidor o archivo, y un cliente o servicio recolector de metadatos. El cliente puede pedir al servidor que le envíe metadatos según determinados criterios, como por ejemplo la fecha de creación de los datos o un rango de fechas. En respuesta el servidor devuelve un conjunto de registros en formato XML, incluyendo identificadores de los objetos descritos en cada registro. En caso de que haya algún tipo de error en la petición (falta de algún argumento, error léxico,..) el servidor devolverá la excepción adecuada.

El protocolo OAI-PMH se compone de seis tipos de peticiones o verbos. Estas peticiones se realizan utilizando los métodos *GET* o *POST* del protocolo http y constan de una lista de opciones con la forma de pares del tipo: clave=valor, como se muestra en la figura 8:

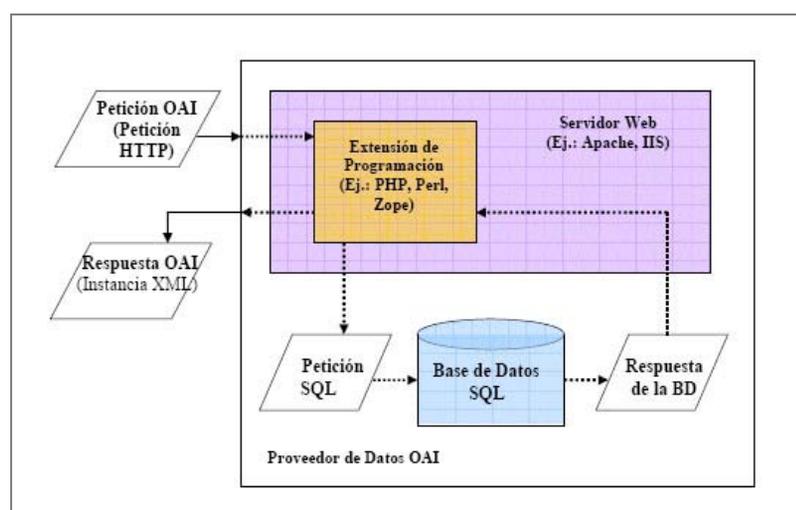


Figura 8: Implementación del Protocolo OAI-PMH [ECH04]

<sup>21</sup> OAI-PMH siglas en inglés de *Open Archives Initiative – Protocol for Metadata Harvesting*.

<sup>22</sup> HTTP (*Hyper Text Transfer Protocol*) es un protocolo de transferencia de hipertexto.

Las seis peticiones que un cliente puede realizar a un servidor son las siguientes:

1. ***GetRecord***: Es utilizado para recuperar un registro concreto. Necesita dos argumentos, el primero es el identificador del registro pedido y el segundo es la especificación del formato bibliográfico en que se debe devolver.
2. ***Identify***: Es utilizado para recuperar información sobre el servidor: nombre, versión del protocolo que utiliza, dirección del administrador, etc.
3. ***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.
4. ***ListRecords***: Recupera registros de un repositorio. Los parámetros posibles son los mismos que para el verbo *ListIdentifiers*.
5. ***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 sólo los registros pertenecientes a una determinada clase. Los conjuntos pueden ser simples listas o estructuras jerárquicas.
6. ***ListMetadataFormats***: Devuelve la lista de formatos bibliográficos que utiliza el servidor. [BAR02]

En la figura 9 se muestra la interacción de un usuario con los distintos repositorios digitales, usando las peticiones del protocolo OAI-PMH.

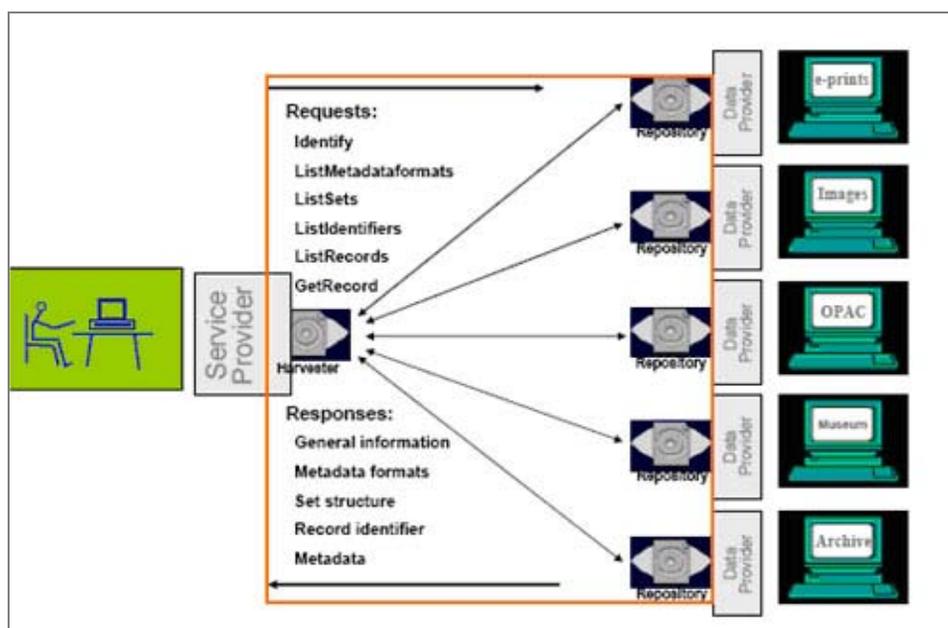


Figura 9: Modelo de Estructura del Protocolo OAI-PMH [OAFnd]

El protocolo 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. Además de éste formato cada servidor es libre de ofrecer los registros en otro(s) formatos adicionales.

Por lo tanto, en vista de las ventajas que ofrece este protocolo, mencionadas anteriormente, éste protocolo es el más utilizado actualmente para la implementación de repositorios digitales. [OAInd] [OAFnd]

## 2.5 Herramientas de Software Libre para la Implementación de Repositorios Digitales.

En la actualidad la disponibilidad de herramientas de software libre para la implementación de repositorios digitales no es elevada, si se compara con otros ámbitos, como la programación, multimedia o herramientas para la creación de portales Web.

Las herramientas de software más importantes para la implementación de repositorios digitales son:

- CDSware
- E-Prints
- Dspace
- Fedora

### 2.5.1 CDSware<sup>23</sup>

El CDSware, ahora llamado CDS Invenio ha sido desarrollado por el CERN<sup>24</sup>. Su objetivo es permitir a cualquier institución la creación de su propio servidor de eprints, catálogos o repositorios digitales a través de la Web.

CDSware utiliza el formato MARC 21 para almacenar los registros bibliográficos, además es un software gratuito bajo la licencia GPL<sup>25</sup>.

CDSware es capaz de soportar gran cantidad de colecciones, y puede ofrecer diferentes portales para cada una de ellas. Integra un potente motor de búsqueda, y permite la personalización de servicios para usuarios registrados. Además puede exportar la información a formatos HTML, XML, MARC, e incorpora el estándar de interoperabilidad OAI. [CDSnd]

### 2.5.2 Dspace

Es un software de código abierto desarrollado por las Bibliotecas del MIT<sup>26</sup> y Hewlett Packard.

Dspace es un repositorio digital, pensado para almacenar, indexar el contenido y difundir los trabajos de investigación de una organización. Este sistema incorpora Dublin Core como estándar de metadatos y OAI-PMH como protocolo de interoperabilidad. Además Dspace está escrito en JAVA<sup>27</sup>, y utiliza las siguientes herramientas:

- JSDK de sun.
- Servidor Web: Apache Webserver.
- Base de datos: Postgres u Oracle.
- Servidor de Páginas Dinámicas: Jakarta Tomcat.
- Motor de Búsqueda Texto Completo: *Lucene search engine*

En Dspace los documentos se organizan en comunidades y sub-comunidades que corresponden a las partes de la organización, tales como departamentos, laboratorios, facultades o escuelas.

La arquitectura modular de Dspace permite la extensión de colecciones multidisciplinarias así como poner límites institucionales, tales como políticas de acceso. Este sistema trabaja con Objetos Digitales, que “Encapsulan atómicamente”, los documentos del usuario, los metadatos agrupados, y los identificadores digitales. Estos

---

<sup>23</sup> CDSware siglas en inglés de CERN *Document Server Software*

<sup>24</sup> CERN siglas en inglés de *European Organization for Nuclear Research*.

<sup>25</sup> GPL siglas en inglés de *General Public License*.

<sup>26</sup> MIT siglas en inglés de *Massachusetts Institute of Technology*

<sup>27</sup> JAVA Lenguaje de programación orientado a objeto.

son indexados para permitir al sistema visualizar colecciones similares y buscar documentos independientes. [DSP]

### 2.5.3 E-Prints

Es un repositorio digital de libre acceso, ha sido desarrollado en el marco del proyecto *Open Access*, por la *School of Electronics and Computer Science* en la Universidad de Southampton.

Una característica de E-Prints es que no incorpora motor de búsqueda propio, ya que está especialmente pensado para interactuar con otros servidores gracias a su potente integración OAI. [EPR]

### 2.5.4 Fedora

Esta basado en *Flexible Extensible Digital Object and Repository Architecture* (FEDORA), fue creada por la Universidad de Cornell y la Biblioteca de la Universidad de Virginia, está desarrollada en JAVA, por lo que necesita el J2SDK.

El concepto básico de Fedora es trabajar con objetos digitales, que poseen un ciclo de vida y unas relaciones entre ellos, a lo largo de un periodo de tiempo. Además ofrece una arquitectura completamente modular y ofrece servicios Web.

Fedora utiliza como estándar de metadatos Dublín Core, y puede aplicar RDF para determinar las relaciones entre los diferentes contenidos, este sistema utiliza como estándar de interoperabilidad el protocolo OAI-PMH.

Una de las principales características de Fedora es que no tiene su propio *Front-End*, por lo cual, se ha desarrollado una interfaz Web llamada Fez, la cual es actualmente desarrollada por la Universidad de Queensland en Australia, como parte de la iniciativa gubernamental del ministerio de educación australiano llamado APSR.

Fez es una interfaz Web Flexible y altamente configurable para el manejo de un repositorio digital y *Workflow* basado en la plataforma FEDORA 2, orientado a distintos tipos de contenidos, comunidades y colecciones.

Las principales características de Fedora son:

- Acceso Web a repositorio y motor de búsqueda avanzada.
- Registro de usuarios, comunidades y colecciones.
- Sólidos protocolos de Seguridad basada en XACML<sup>28</sup>(FezACML)
- Roles: *Creator, Editor, Approver, Viewer, Lister*.

---

<sup>28</sup> XACML siglas en inglés de *eXtensible Access Control Markup Language* (Lenguaje de marcas para el Control extensible de acceso).

- Flexible manejo de tipos de archivo (MIME<sup>29</sup> XML *Type*)
- Metadatos basado en Dublín Core.
- El sistema *workflow* administra el ciclo de vida de un objeto, desde su creación, aprobación, edición y publicación.

[FED]

---

<sup>29</sup> MIME siglas en inglés de *Multipurpose Internet Mail Extensions* (Extensiones de Correo Internet Multipropósito)

### 3. FEDORA/FEZ

Fedora entre los distintos sistemas de gestión documental, se destaca por su gran potencial de flexibilidad y personalización de los modelos de contenido, además de su sólido esquema de objetos XML.

#### 3.1 Fedora (*Back-end*)

El sistema de gestión documental **FEDORA** es un sistema para repositorios digitales basados en estándares abiertos (XML, PHP, MySQL). Puede ser usado como arquitectura base, para cualquier repositorio institucional y biblioteca digital basada en tecnología Web.

Fedora utiliza el protocolo de interoperabilidad OAI-PMH, lo cuál significa que sus contenidos pueden incorporarse a redes de colecciones similares de todo el mundo y ser buscados por los agentes de Internet, como Google, Google Scholar, o servicios OAI como Oaister.

##### 3.1.1 Modelo Digital de Fedora

El modelo del objeto digital de FEDORA es como un contenedor que estructura los datos que contiene, en donde la agregación de elementos, puede utilizarse individualmente o como parte de un conjunto.

Fedora se basa en la idea de que los objetos digitales pueden ser:

- **Heterogéneos:** Fedora define un modelo de objetos digital genérico que puede representar diferentes tipos de objetos como: objetos textuales, imágenes, libros electrónicos, objetos multimedia, *datasets*, metadatos y muchos otros objetos.
- **Complejos:** Fedora soporta la agregación de uno o más ítems de contenidos en un mismo objeto digital (objetos digitales compuestos, como por ejemplo un artículo que contiene una parte textual, gráficos y metadatos). El contenido puede ser de cualquier tipo de formato, y puede estar almacenado en el repositorio localmente o fuera del repositorio (en servidores externos) referenciado por el objeto digital.
- **Generados dinámicamente:** Fedora ofrece mecanismos de acceso al objeto digital que permiten transformar dinámicamente el contenido para un uso

específico o un contexto determinado. (Ej: Ofrecer una vista ampliada de una imagen o sólo la tabla de contenidos del documento). Estos mecanismos se denominan en el sistema “*disseminators*”, el cuál es un componente opcional utilizado para extender los puntos de acceso al objeto digital. El “*disseminator*” apunta a un conjunto de métodos que son invocados para producir representaciones virtuales del objeto. Una representación virtual es cuándo un contenido que no está explícitamente almacenado en un objeto digital, sino que se produce dinámicamente en el momento en que es invocado el servicio.

- **Relacionados:** Los objetos digitales en Fedora pueden contener metadatos que expresen sus relaciones jerárquicas, relaciones asociativas como “Es miembro de” (por ejemplo una tesis es miembro de una colección llamada “Tesis de la Escuela Ingeniería Civil en Informática”), relaciones estructurales del tipo “Es parte de” (artículo/revista, capítulo/libro). Estas relaciones se expresan con la ontología RDF y emplean una base de datos relacional para su almacenamiento y disponen de un lenguaje específico de consulta.

El sistema soporta tres tipos de metadatos:

### **1. Metadatos administrativos**

Son aquellos metadatos que describen la historia, los derechos de acceso, las versiones, etc. De cada objeto digital, son creados por el propio sistema. Estos metadatos permiten crear índices para las tareas de gestión interna del repositorio.

### **2. Metadatos técnicos**

Describen los formatos y características técnicas del objeto, son opcionales, éstos sirven para crear índices tanto para las tareas de gestión como para las tareas del usuario final.

### **3. Metadatos descriptivos**

Describen el contenido intelectual del objeto. Son los metadatos que tradicionalmente crea la biblioteca para describir sus colecciones (autor, título, etc.). Sirven para crear los índices que permitirán la recuperación del objeto (similar a lo que sucede con los catálogos).

FOXML es el esquema XML que representa al objeto digital de Fedora, donde la estructura del objeto digital de Fedora se muestra en la figura 10 y 11:

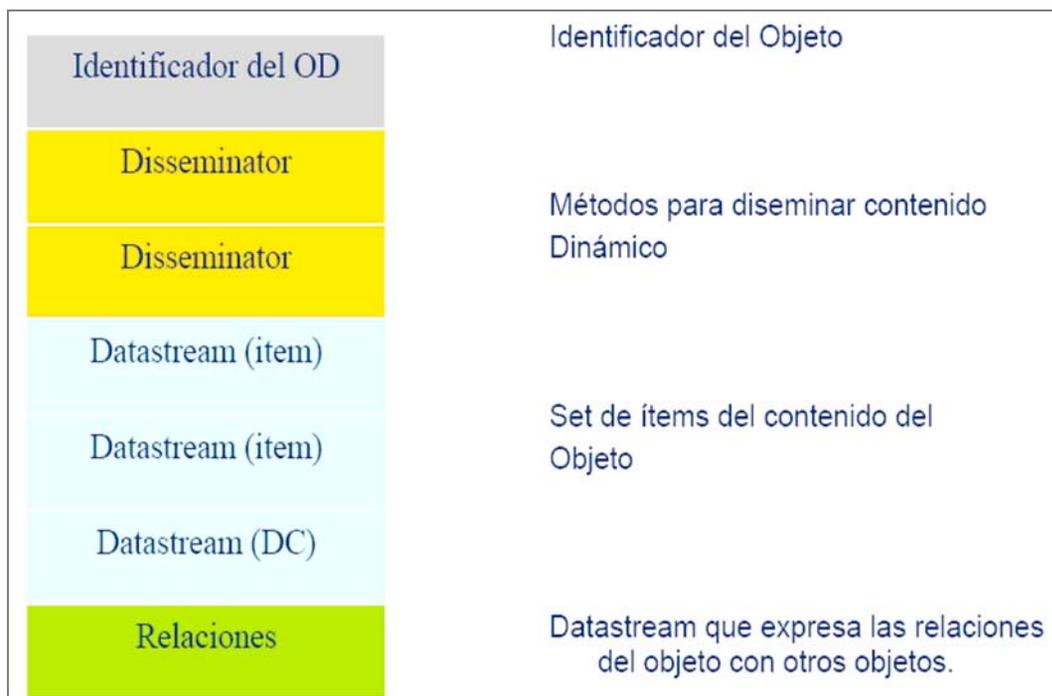


Figura 10: Estructura del Objeto Digital de Fedora.

```

<digitalObject PID="uniqueID">
  <!-- there are a set of core object properties -->
  <objectProperties>
    <property/>
    <property/>
    ...
  </objectProperties>
  <!-- there can be zero or more datastreams -->
  <datastream>
    <datastreamVersion/>
    <datastreamVersion/>
    ...
  </datastream>
  <!-- there can be zero or more disseminators -->
  <disseminator>
    <disseminatorVersion/>
    <disseminatorVersion/>
    ...
  </disseminator>
</digitalObject>

```

Figura 11: Esquema del objeto digital de Fedora (FOXML).

En el transcurso de revisión para este proyecto de tesis, se encontró un documento de la UNED (Universidad Nacional de Educación a Distancia) [UNE05] donde se rescato importante información sobre FEDORA. El cuál se ve reflejado en los siguientes puntos (3.1.2, 3.1.3, 3.1.4).

### 3.1.2 Arquitectura de los Servicios Fedora

Todas las funciones de Fedora, tanto a nivel del objeto como del repositorio digital, se exponen como servicios Web. El marco del servicio de Fedora está hecho para facilitar la integración de nuevos servicios y funcionalidades sobre el repositorio digital Fedora. Mientras que el repositorio en sí mismo expone sus funcionalidades como un conjunto de interfaces de servicios Web, todas estas interfaces pertenecen a la aplicación Web de Fedora que corre en su propio Tomcat.

El marco de servicios de Fedora permite construir nuevos servicios en torno al núcleo del repositorio como aplicaciones Web “*stand-alone*” que corren independientemente del repositorio Fedora. Esta aproximación aporta dos ventajas fundamentales:

1. Permite que se añadan nuevas funcionalidades como servicios modulares que pueden interactuar con el repositorio Fedora y sin embargo no forman parte de este repositorio.
2. Facilita el co-desarrollo de nuevos servicios Fedora, dado que cada servicio puede ser desarrollado y conectado de forma independiente a la arquitectura central de Fedora, como se muestra en la figura 12.

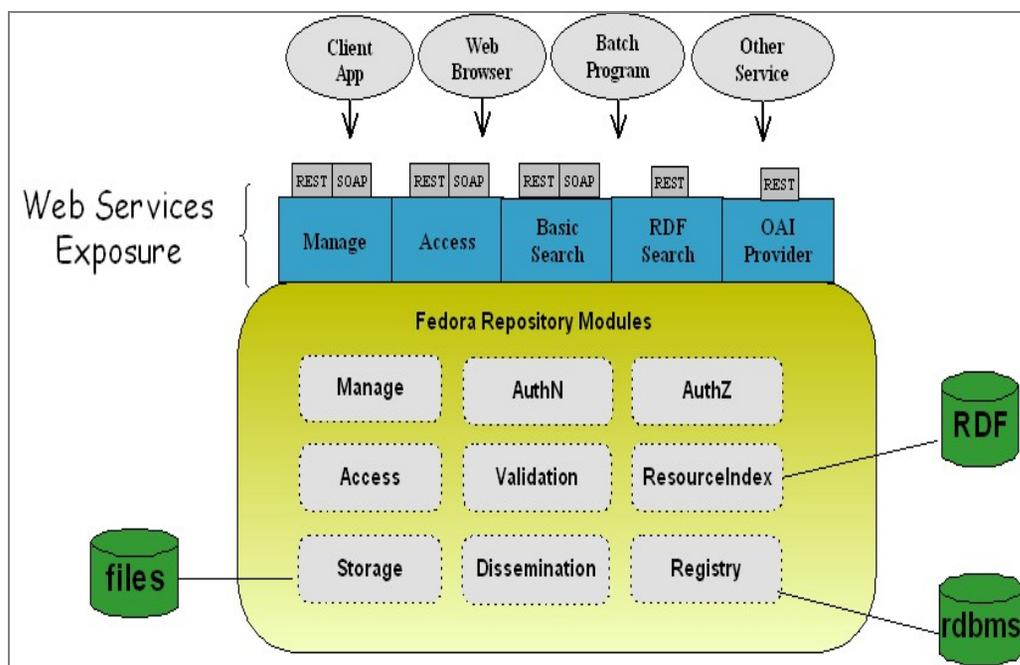


Figura 12: Arquitectura de Fedora [PAY05]

La arquitectura de Fedora no está construida para ningún *workflow* o aplicación de usuario final en particular. Por ello, Fedora puede funcionar como el substrato de un repositorio genérico sobre el cuál se pueden crear muchos tipos de aplicaciones y

aprovechar las ventajas, en el futuro de los desarrollos en servicios Web. La capacidad de crecer y cambiar garantiza la longevidad de la arquitectura del modelo digital Fedora.

A continuación se muestran dos figuras, en una (Figura 13) se muestra la arquitectura principal de Fedora, y en la otra (Figura 14) se muestra los servicios adicionales que pueden ser creados gracias a la flexibilidad de la arquitectura de Fedora. [UNE05]

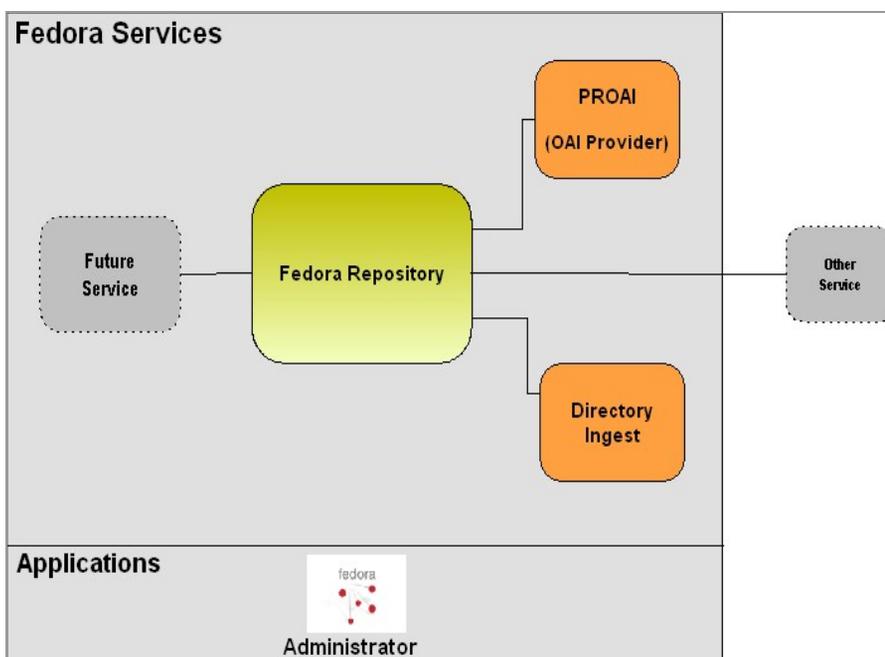


Figura 13: Arquitectura Principal de Servicios de Fedora [PAY05]

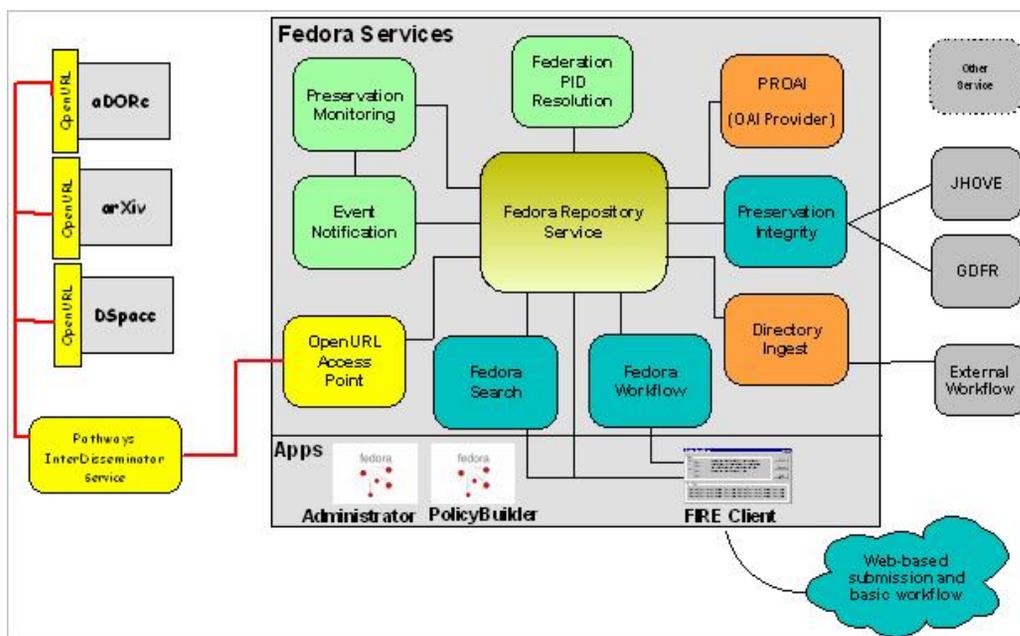


Figura 14: Arquitectura de Servicios de Fedora [PAY05]

### 3.1.3 Archivo y Preservación de los Objetos

Las capacidades de archivo y preservación de Fedora incluyen:

- **XML:** Los objetos XML de Fedora y el esquema sobre el que se basan son preservados durante su carga, almacenamiento y exportación.
- **Control de versiones del contenido:** Fedora permite seguir los cambios y modificaciones de un objeto (el cambio, quién y cuándo se hizo) añadiendo la nueva versión al objeto XML. Este nuevo ítem se añade en cascada al objeto digital y se numera para mostrar la relación entre el original y la versión. Ello permite recuperar antiguas versiones del objeto realizando una búsqueda por fecha/hora o la versión más actualizada, si no se incluye el criterio de fecha en la búsqueda.
- **Histórico:** Cada objeto del repositorio Fedora contiene una auditoria que preserva un registro de cada cambio realizado sobre el objeto. [UNE05]

### 3.1.4 Autenticación/Organización

La versión de Fedora 2.1.1 incluye soporte para la autenticación, SSL<sup>30</sup>, y autorización usando políticas XACML. Estas medidas de seguridad son gestionadas a través de un conjunto de políticas de XACML y con información de autenticación a través de Tomcat o fuentes externas de autenticación. [UNE05]

## 3.2 Fez (*Front-end*)

Para el diseño de Fez se utilizó la metodología orientada a objeto y se separó la lógica de negocio con la presentación del sistema, utilizando las plantillas Smarty. [SMRnd]

### 3.2.1 Modelo de Contenido.

Fez está diseñado en torno a la idea de utilizar capas jerárquicas, las cuáles son las comunidades, colecciones y los registros. Las comunidades contienen varias colecciones y cada colección contiene varios registros. Estas relaciones son asignadas por el estándar de datos de Fedora RELS-EXT.

---

<sup>30</sup> SSL siglas en inglés de Secure Sockets Layer (Seguridad de la Capa de Transporte).

Una de las principales desventajas en el sistema de gestión documental e-Print es la dificultad existente para la personalización de las plantillas y la creación de nuevas plantillas para poder gestionar todo tipo de documentos, ya que cada tipo de documentos tiene una especificación de requisitos distinta. (Ej: imágenes, revistas, documentos de Tesis ..., etc).

Fez puede manejar cualquier modelo de contenido, utilizando los objetos XSD<sup>31</sup>.

**XML Schema** es 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.

“XML Schema” (Esquema XML) es el nombre oficial otorgado por la W3C, pero se recomienda utilizar el término “Definición de Esquema XML”, y reservar “XML Schema” para la denominación del lenguaje específico. [W3C04b]

Los esquemas XML (XSD) se presentan al usuario en forma de formulario Web HTML lo que permite un alto grado de configuración en la creación de plantillas de interfaz Web, por lo tanto, los registros pueden tener distintos tipos de plantillas, para poder gestionar todo tipo de documentos, los cuáles son definidos y configurados en las interfaces administrativas de Fez.

### 3.2.2 Seguridad – FezACML

FezACML fue diseñado para resolver las urgentes necesidades de seguridad, su diseño esta basado en la creación de roles. Estos roles se otorgan a un usuario cuando se satisface una de las condiciones del par Condición-Roles.

Los roles utilizados por Fez son:

**Lister:** Este rol es para que los usuarios puedan ver un objeto en los listados y resultados de búsqueda. Por defecto todos los usuarios pueden ver los objetos en los listados y resultados de búsqueda.

**Viewer:** Este rol permite que los usuarios vean los objetos. Por defecto todos los usuarios pueden ver los objetos, además si a un usuario se le asigna este rol sobre un objeto, por defecto al usuario se le asignará el rol *Lister*.

**Creator:** Este rol permite a los usuarios crear objetos sobre un objeto determinado. Si un usuario tiene asignado el rol *Creator*, por defecto se le asigna el rol *Viewer*.

---

<sup>31</sup> XSD siglas en inglés de XML Schema Definition (Definición de Esquema xml).

Ej: Un usuario tiene el rol *Creator* sobre una comunidad, esto permite al usuario poder crear colecciones dentro de dicha comunidad.

***Editor***: Este rol permite a los usuarios poder editar un objeto y a sus objetos hijos. Si un usuario tiene asignado el rol *Editor*, por defecto se le asigna el rol *Viewer*.

***Approver***: Este rol permite a los usuarios poder publicar (aprobar) archivos de un determinado objeto, que están en espera de ser publicados.

Sólo a los usuarios que se encuentran registrados en el sistema, se les puede asignar roles.

El conjunto de reglas de seguridad para los objetos Fez/Fedora son guardados en su propio *datastream* llamado FezACML.

FezACML está diseñado para que la seguridad pueda ser heredada jerárquicamente por los objetos padres. Esto permite que las reglas de toda una comunidad o colección se establezcan en un solo lugar. [KOR06]

### **3.2.3 Workflow**

Además de la flexibilidad y seguridad de los documentos, Fez está diseñado para conseguir una gran flexibilidad y fácil personalización de la interfaz, para el control del flujo de la GUI<sup>32</sup> y la administración de documentos.

El diseño del *workflow* de Fez está basado en estados, *triggers* y comportamientos, como se muestra en la figura 15.

---

<sup>32</sup> Interfaz gráfica del usuario.

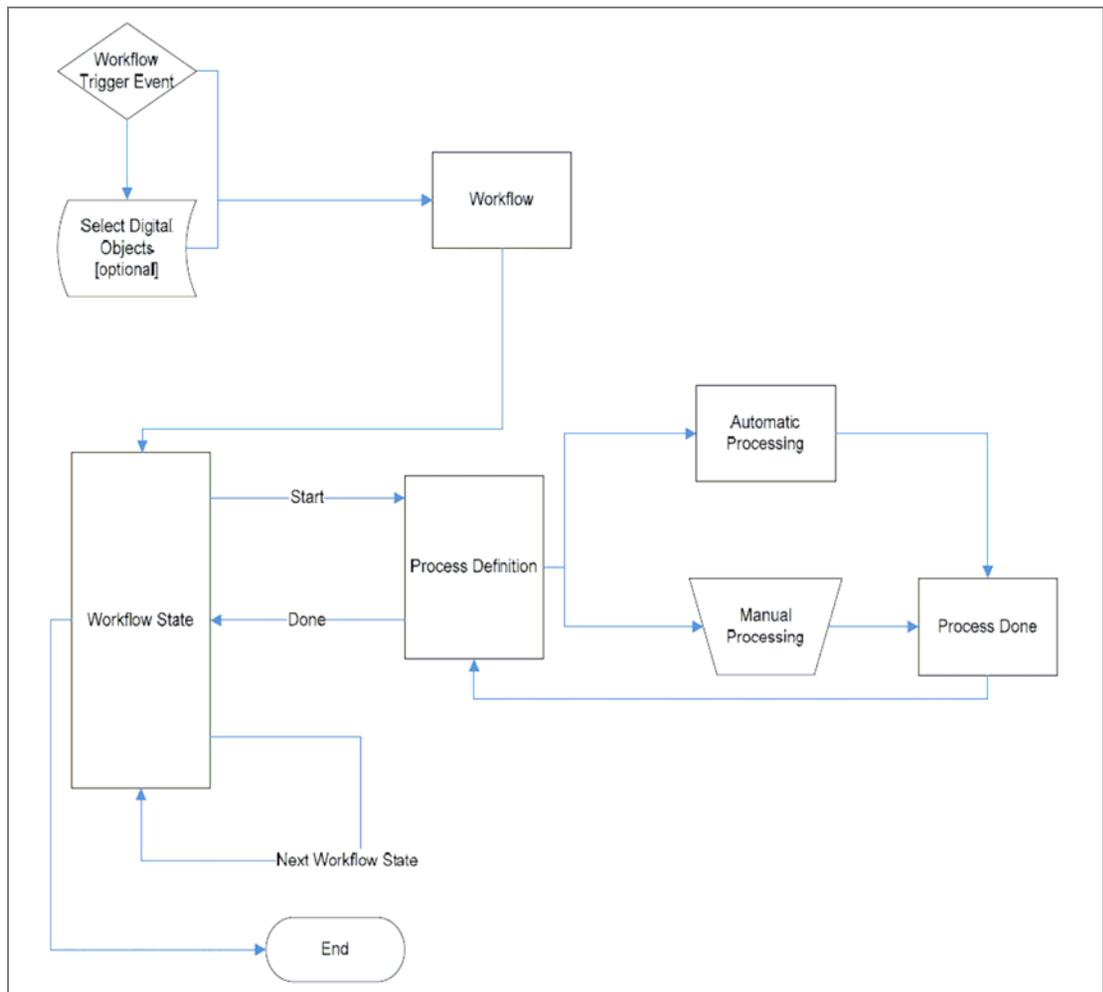


Figura 15: Diagrama de flujo de los componentes del workflow de Fez.

Los *Workflows* son sistemas basados en estados, donde cada estado tiene asociado un proceso. La definición del proceso puede ser automática o manual.

Al finalizar el proceso de un estado, el *workflow* pasa al siguiente estado. Cada estado tiene asociada una acción y puede tener privilegios de acceso basados en las restricciones del rol de usuario FezACML.

Por defecto un conjunto de *triggers* se aplican a todos los objetos de Fez, donde la personalización del *workflow* puede ser definido en cualquier nivel jerárquico del objeto, similar a la seguridad FezACML.

El panel de administración de *workflow* en Fez proporciona una forma fácil de definir y comprender este complejo sistema. Un ejemplo del *workflow* de administración de interfaz de la “creación de registro” se puede ver en la figura 16:

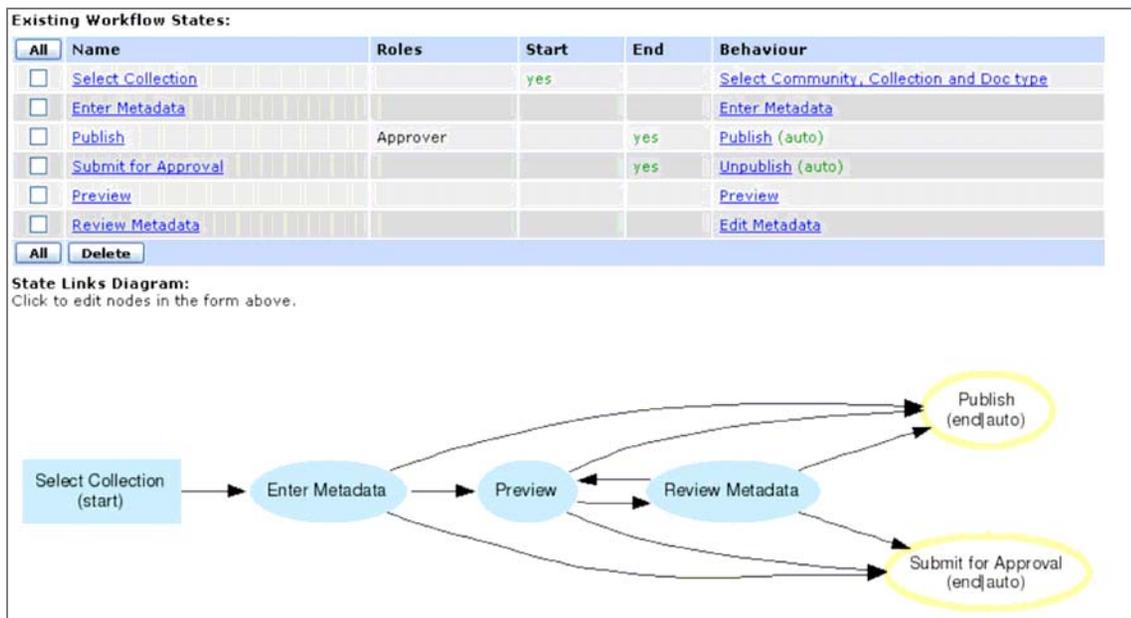


Figura 16: Diagrama de estados del *workflow* para la creación de registros.

El diagrama de la figura 15 es generado automáticamente con *Graphviz* y está basado en las reglas de estado del *workflow*. El usuario puede hacer “click” en los nodos del diagrama, para cargar el enlace correspondiente al estado y así poder editarlo. [KOR06]

### 3.2.4 Índice de Búsqueda.

Uno de los principales problemas encontrados con el software Fedora, fue su servicio de búsqueda, ya que esta es muy básica. El servicio de búsqueda de Fedora sólo permite la búsqueda de datos a través de los elementos del estándar Dublín Core, no provee una lógica booleana, y tiene una API limitada.

Debido a estas limitaciones Fez tiene su propio índice y sistema de búsqueda, para ello utiliza el poder de búsqueda booleana de MySQL.

En “Administrador” -> “*Manage Search Keys*” es dónde se administra los índices de búsqueda en Fez, aquí se ingresan las distintas clasificaciones de índices de búsqueda que serán indexados con los atributos (Ej: Título, Autor, Patrocinante, Tema, etc.). En éste panel se especifica para cada índice que tipo de campo es (*text input, combo box, text area, controlled vocabulary, etc.*), además de especificar en que tipo de búsqueda se mostrará el campo. Por ejemplo especificar si el índice título se mostrará en la búsqueda avanzada o en la búsqueda básica, o en ambas. Como se muestra en la figura 17, 18 y 19.

**Manage Search Keys**

**Title:**

**Alternative Title (if empty Title will be used, useful for language translations of Fez):**

**Use in Simple Search?:**  (If this tickbox is enabled any fields using this search key will be included in a simple search)

**Show in Advanced Search Form?:**  (If this tickbox is enabled this search key will show up as an option checkbox in the advanced search form)

**Show in My Fez Search Form?:**  (If this tickbox is enabled this search key will show up as an option checkbox in the My Fez search form)

**Display Order:**

**Multiple?:**  (If this enabled tickbox is ticked this element will be able to be entered multiple times in forms)

**Multiple Limit:**  (The maximum amount of html input elements that will show for this element if it is a multiple)

**Field Type:**  Text Input  Text Area  Combo Box  Multiple Combo Box  
 Check Box  Date Selector  Controlled Vocabulary  All Controlled\_vocabularies

**Function Lookup for Value of ID (optional):**

Figura 17: Ingreso de una índice de búsqueda.

Existing Search Keys:

All	Title	Alt Title	In Simple Search?	On Adv Search Page?	On MyFez Search Page?	Multiple?	Multiple Limit	Field Type	Order
<input type="checkbox"/>	Title	Título	yes	yes	yes	no		text	0

Figura 18: Índice de búsqueda Ingresado.

Al crear un objeto XSD, el “*XSD HTML Matching Editor*” permite asignar un índice de búsqueda (*search key*) a los campos del objeto (como por ejemplo título, autor, patrocinante). Esto permite a los usuario realizar búsquedas por cualquier campo, independiente del estándar que utilice el objeto (dublín core, mods, rdf).

<b>Title:</b>	<input type="text" value="Titulo"/>
<b>Label Description:</b>	<input type="text"/> (it will show up by the side of the label)
<b>Field Description:</b>	<div style="border: 1px solid gray; height: 100px; width: 100%;"></div> (it will show up by the side of the field)
<b>Image Location:</b>	<input type="text"/> (It will be shown to the left of the title of the element if it exists in the images directory of Fez.)
<b>Enforced Namespace Prefix:</b>	<input type="text"/> (it will be used as the namespace for this field and override the main object namespace. You must include the ":".)
<b>Value Prefix:</b>	<input type="text"/> (it will be appended to the front of any form entered variable for this xsdmf match eg for Pids to have info:fedora/ at the front.)
<b>Display Order Priority:</b>	<input type="text" value="1"/> (The order the field will display on HTML forms, 0 is highest priority, 1, 2 next etc.)
<b>Enabled in Form?:</b>	<input checked="" type="checkbox"/> This should be checked to make this field appear on the forms for editing metadata. If the field is static text, it will only be shown if the 'Show in View Details' box is also checked (used to make a section heading in the form) . If this box is unchecked, the resulting XML will have a blank value for this element.
<b>Indexed?:</b>	<input checked="" type="checkbox"/> (If this indexed tickbox is ticked data entered into this xml element will be used in searches in the Fez index)
<b>Search Key:</b>	<input type="text" value="Titulo"/>

Figura 19: Indexación del campo “Titulo” al índice de búsqueda “Titulo”.

Una de las ventajas de Fez es que si un índice o todos éstos son corrompidos se puede regenerar por una re-indexación del repositorio Fedora (*back-end*). Esto se logra mediante el grupo de administración de Fez “*Index Fedora Object into Fez*” y “*Reindex Fez Fedora Objects*”. [FEZnd]

### 3.2.5 Índices de Seguridad.

Para aumentar el rendimiento de las consultas a las reglas de seguridad de FezACML, se creó un índice de seguridad, que fue diseñado similarmente al índice de búsqueda de Fez. Un rediseño del índice de seguridad se realizará en la próxima versión de Fez, donde se almacenará de forma separada los pares de reglas, de las reglas de PID y la combinación del ID *datastream*. Esto hará al índice de seguridad mucho más eficiente.

El quiebre de la relación entre la regla y el objeto que estarán en tablas separadas, permitirá a MySQL acelerar las consultas al índice dramáticamente. [FEZ]

### 3.2.6 Control de Vocabulario

El control de Vocabulario es capaz de clasificar objetos según diferentes clasificaciones sustanciales, dichas clasificaciones no sólo pueden ser texto, sino que también se puede asociar imágenes a los temas de clasificación, existen 2 tipos de clasificaciones:

1. Clasificaciones Planas, como por ejemplo un control de vocabulario que contenga los nombres de todas las ciudades de Chile.
2. Clasificaciones Jerárquicas, como por ejemplo crea un control de vocabulario que contenga todas las facultades de la Universidad Austral de Chile, y dichas facultades tengan todas las escuelas asociadas a ellas.

### 3.3 Arquitectura del Repositorio Digital Fedora/Fez

La arquitectura de software general del Repositorio Digital es MVC<sup>33</sup>. El Modelo Vista Controlador es un patrón de arquitectura de software que separa los datos de una aplicación, interfaz de usuario, y de la lógica de control entre sus distintos componentes.

**Modelo:** Es la representación específica de la información con la cuál el sistema opera. La lógica de datos asegura la integridad de éstos y permite derivar nuevos datos.

**Vista:** Presenta el modelo en un formato adecuado para que el usuario interactúe con el sistema.

**Controlador:** Responde a eventos, usualmente acciones del usuario donde invoca cambios en el modelo y probablemente en la vista. En una aplicación Web el controlador recibe la petición del usuario, interactúa con el modelo para procesar los datos y hace disponible esos datos a la vista.

En el repositorio digital la arquitectura MVC se distribuye de la siguiente manera:

**Modelo:** Se encuentran en una carpeta llamada “*include*” la cuál contiene todas las clases que interactúan con la base de datos.

**Vista:** Se encuentran en una carpeta llamada “*Template*” en la cuál están todas las páginas HTML que interactúan con el usuario.

**Controlador:** Estas no se encuentran en una carpeta específica, son las que interactúan con los archivos HTML y con las clases del sistema.

Ej. Al ingresar un nuevo usuario al sistema, el usuario ingresa los datos a la página HTML (Vista) llamada “*user.tlp.html*”, la cuál llama al archivo *user.php* (Controlador), donde se llama a la clase “*class.user.php*” (Modelo) para ingresar los datos a la base de

---

<sup>33</sup> Modelo Vista Controlador

datos, luego responde que los datos fueron ingresados con éxito al user.php, el cuál lo representa en la vista para que el usuario lo vea, como se muestra en la figura 20:

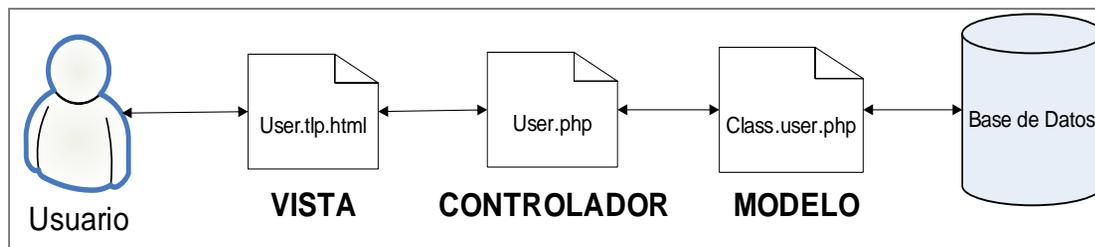


Figura 20: Esquema de Modelo-Vista-Controlador.

En la figura 21 se muestra un diagrama del repositorio digital Fedora/Fez, dónde se muestra la relación existente entre sus servicios.

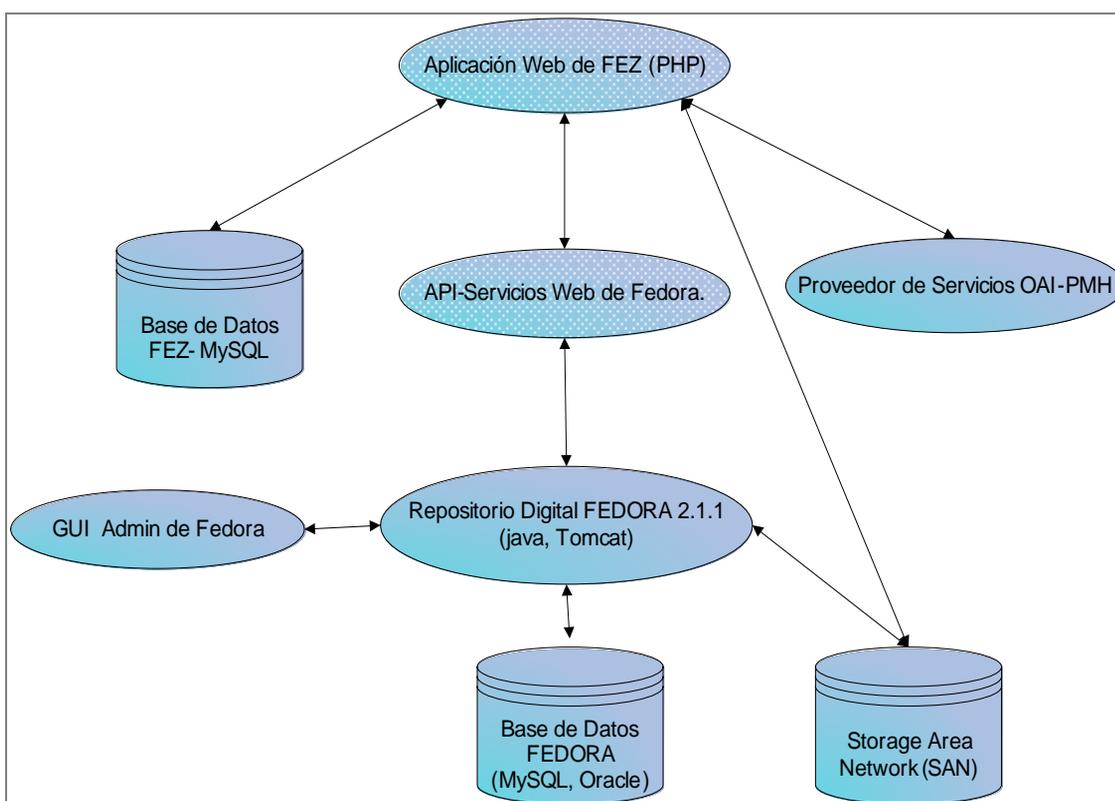


Figura 21: Diagrama de la arquitectura del repositorio Digital Fedora/Fez

## **4. REPOSITORIO DIGITAL EN UNA ORGANIZACIÓN. UACH, FVT.**

### **4.1 Propuesta UACH**

Es posible implementar un repositorio digital robusto, flexible y escalable, para la biblioteca de la Universidad Austral de Chile, para gestionar los documentos de tesis de toda la Universidad, asegurando la recuperación de la información de forma fácil, y permitiendo la reutilización de los contenidos para diferentes propósitos y contextos. Además proporcionará un acceso centralizado a todos los documentos de tesis, que incrementará y mejorará su visibilidad y difusión.

El repositorio digital permitiría ingresar los documentos de tesis vía Web, dónde el repositorio estará compuesto por comunidades que representarán las facultades de la universidad, donde cada comunidad (Facultad) tendrá asociado un conjunto de colecciones que representarán las escuelas de cada facultad, y cada escuela tendrá un conjunto de registros que serán los documentos de tesis de alumnos titulados de la Universidad Austral de Chile.

Cada usuario que esté registrado estará asociado a un grupo de usuarios, donde cada grupo tendrá ciertos privilegios sobre las distintas comunidades (Facultades) y colecciones (Escuelas). Como por ejemplo, todos los grupos de “Alumnos” podrán subir archivos, pero éstos serán revisados por el grupo “Personal de Biblioteca”, antes de ser publicados. Además el grupo “Alumnos de Ingeniería Civil en informática” sólo podrán ingresar archivos a la colección “Escuela de ingeniería Civil en Informática”.

El repositorio tendrá 2 sistemas de búsqueda, uno simple y otro avanzado, además de las búsquedas predeterminadas, las cuáles son: búsqueda por autor, patrocinante, año y por los últimos documentos ingresados.

Una ventaja que poseerá este sistema será que utilizará la base de datos de la organización para la autenticación de los usuarios del sistema, para ello, Fez se conectará a la base de datos de la organización de forma remota.

#### **4.1.1 Metodología de Implementación**

Para implementar el repositorio digital en la Universidad Austral de Chile se utilizará el repositorio digital FEDORA 2.1.1 (*back-end*), y como interfaz Web (*front-end*) Fez 1.3. Para el registro de los documentos de tesis se utilizará el estándar de metadatos MODS y para la interoperabilidad se usará el estándar OAI-PMH.

El protocolo que se utilizará para la comunicación entre el repositorio digital y la base de datos de la organización será XML-RPC.

XML-RPC<sup>34</sup> es un protocolo de llamadas a procedimientos remotos, el cuál trabaja sobre Internet. Un Mensaje XML-RPC es una petición HTTP-POST, dónde el cuerpo de ésta petición es un XML.

Cuando un procedimiento se ejecuta en el servidor, el valor devuelto también está formateando en XML.

Los parámetros del procedimiento pueden ser escalares, números, cadenas de caracteres, datos, etc. Además pueden ser complejas estructuras de registros y listas.

Los pasos para la implementación del repositorio digital serán los siguientes:

1. Instalación de software, librerías y base de datos que son necesarios para la correcta instalación del repositorio digital.
2. Instalación del *back-end* FEDORA 2.1.1.
3. Instalación del *front-end* Fez 1.3.
4. Definir los tipos de documentos que se ingresarán al repositorio digital, y los campos asociados a dichos documentos, que serán representados por los objetos digitales (metadatos).
5. Creación del objeto digital que representará al documento de tesis, utilizando el estándar de metadatos MODS.
6. Creación de los índices de búsqueda, para la posterior indexación con los campos del objeto digital, que representará al documento de tesis (título, autor, patrocinante, etc.) en el repositorio digital.
7. Definir las áreas y sub-áreas de la organización, que serán representadas por las comunidades y colecciones en el repositorio digital.
8. Creación de comunidades y colecciones para la administración de los documentos de tesis.
9. Administración de las políticas de seguridad para las comunidades y colecciones.
10. Creación de búsquedas personalizadas (ej. Buscar por Autor, Patrocinante, Fecha, entre otros datos).
11. Creación de la API para la conexión entre el repositorio digital y la base de datos de una organización para la autenticación de los usuarios.

---

<sup>34</sup> RPC son siglas en inglés de *Remote Procedure Calling* (Llamadas a Procedimientos Remotos).

#### **4.1.2 Trabajo Realizado.**

En el marco del proyecto de título, se realizaron entrevistas y se implementó una maqueta funcional del sistema, se realizaron reuniones y se mostró el funcionamiento del sistema a responsables de la Biblioteca de la Universidad Austral de Chile, se pudo determinar como una muy buena alternativa a los requerimientos de almacenamiento y acceso a documentos.

El sistema sólo se evaluó como una versión de prueba y no se continuó su desarrollo para la Universidad, esto debido a que ya se contaba con otro sistema hecho en la misma Universidad, y por razones de presupuesto, y tiempos de desarrollo e implementación.

Los resultados de la evaluación de la versión de prueba del sistema con personal de la biblioteca de la UACH fueron los siguientes:

- Muy buen nivel de usabilidad del sistema.
- Buen nivel de búsqueda.
- Muy buen nivel de administración de los documentos de tesis, utilizando las comunidades (Facultades) y colecciones (Escuelas).
- Muy buen nivel de administración de la seguridad del sistema, y sus roles de seguridad de cada usuario. Pero para los requerimientos del personal de la biblioteca solo necesitaban un rol (administrador), ya que ellos mismos ingresaban las tesis al sistema. Por lo tanto, en este punto el sistema no se iba a utilizar a su cien por ciento.
- Se encontró de mucha utilidad el despliegue de la información personalizada por Patrocinante, Fecha, Facultad, Escuela y Autor.
- Fácil manejo de noticias relacionadas con las tesis, para su publicación en el sistema Web.

Uno de los puntos interesantes fue el conocimiento que fue compartido por el personal de biblioteca de la universidad para definir conceptos como colecciones, comunidades, definición de campos a utilizar en los metadatos para la representación de los documentos de tesis, y medios de indexación de documentos

## **4.2 Propuesta FVT.**

Similar a lo detallado en el ítem anterior, luego de evaluar y realizar pruebas con un prototipo con personal de la Universidad Austral de Chile, se procedió a evaluar el impacto de la implementación en la empresa FVT Ltda. ([www.fvt.cl](http://www.fvt.cl)), para ello se implantó el sistema en sus dependencias, el cual estuvo instalado por un período de 1 mes.

### **4.2.1 Metodología de Implementación.**

Se siguieron los pasos definidos en 4.1.1. Mayores detalles se indican en el capítulo 6, donde se realiza una evaluación del sistema.

### **4.2.2 Trabajo Realizado.**

Se realizaron reuniones con ingenieros de la empresa FVT Ltda, luego de definir la estructura de almacenamiento de sus documentos (los relevantes para las pruebas), se procedió a realizar la instalación física de un servidor en sus dependencias.

## 5. IMPLEMENTACIÓN DEL REPOSITORIO DIGITAL FEDORA/FEZ

En este capítulo se explicará los pasos a seguir para implementar el repositorio digital, los cuales son:

### 5.1 Guía de Instalación de Repositorio Digital Fedora 2.1.1 y su Front-end Fez 1.3, en Sistema Operativo Fedora Core 7

En ésta sección se especificará los pasos a realizar para instalar de forma correcta el sistema de Repositorio Digital Fedora/Fez.

#### 5.1.1 Requisitos

Los programas necesarios para poder instalar de forma correcta el repositorio digital, se pueden ver en la tabla 3.

Tabla 3: Software necesarios para el Repositorio Digital

Programas	Dirección Web para Descargar
Servidor Apache	Se instala desde el sistema operativo.
PHP 5	<a href="http://www.php.net/downloads.php">http://www.php.net/downloads.php</a>
Base de datos MySQL	<a href="http://dev.mysql.com/downloads/">http://dev.mysql.com/downloads/</a>
phpMyAdmin	Se instala desde el sistema operativo.
Java J2SDK	<a href="http://java.sun.com/j2se/1.4.2/download.html">http://java.sun.com/j2se/1.4.2/download.html</a>
Librería libpng	<a href="http://www.libpng.org/pub/png/libpng.html">http://www.libpng.org/pub/png/libpng.html</a>
Librería libjpeg	<a href="http://www.ijg.org/files/">http://www.ijg.org/files/</a>
Librería CURL	<a href="http://curl.haxx.se/latest.cgi?curl=tar.gz">http://curl.haxx.se/latest.cgi?curl=tar.gz</a>
Librería Tidy	<a href="http://tidy.sourceforge.net/">http://tidy.sourceforge.net/</a>
ImageMagick	<a href="ftp://mirror.aarnet.edu.au/pub/ImageMagick">ftp://mirror.aarnet.edu.au/pub/ImageMagick</a>
Graphviz	<a href="http://www.graphviz.org/pub/graphviz/ARCHIVE/">http://www.graphviz.org/pub/graphviz/ARCHIVE/</a>
JHOVE	<a href="http://hul.harvard.edu/jhove/distribution.html">http://hul.harvard.edu/jhove/distribution.html</a>
Librería domxml	<a href="http://es2.php.net/downloads.php">http://es2.php.net/downloads.php</a>
Server Fedora-2.1.1	<a href="http://www.Fedora.info/download/2.1/download.cgi">http://www.Fedora.info/download/2.1/download.cgi</a>
Fez 1.3	<a href="http://sourceforge.net/project/showfiles.php?group_id=148409">http://sourceforge.net/project/showfiles.php?group_id=148409</a>

### 5.1.2 Actualización de Paquetes en Fedora 7.

Lo primero que se debe hacer es actualizar los paquetes del Sistema Operativo Fedora core 7, para ésto, se debe obtener los privilegios del Usuario *Root*, ejecutando el siguiente comando en la consola:

```
su  
(Contraseña de usuario root)
```

Luego para Actualizar los paquetes necesarios en Fedora se debe ejecutar el siguiente comando en la consola:

```
Yum -y update && reboot
```

Donde **yum**<sup>35</sup> es el manejador de paquetes por excelencia de Fedora, este comando actualiza todos los paquetes que tengan actualizaciones en el sistema, la opción `-y` hace que la descarga y actualizaciones se realicen automáticamente sin que tengamos que intervenir, el comando **reboot** lo que hace, es reiniciar el sistema después que termina la actualización.

### 5.1.3 Instalación del Servidor Apache

Para instalar el servidor Apache 2 se debe ejecutar el siguiente comando en la consola:

```
Yum install httpd
```

Luego se debe iniciar el servicio del servidor Apache ejecutando el siguiente comando en la consola:

```
service httpd start
```

Los archivos de configuración de Apache están ubicados en `/etc/httpd/conf/httpd.conf`

---

<sup>35</sup> Yum son siglas en inglés de Yellow Update Manager (Gestor de Actualizaciones de Fedora).

### 5.1.4 Instalación de PHP 5

Para instalar PHP 5 se debe ejecutar el siguiente comando en la consola:

```
Yum install php
```

Esto instalará la última versión de php.

La carpeta Web por defecto está ubicada en /var/www/html/

Una vez instalado php se debe reiniciar el servidor Apache para que los cambios hagan efecto:

```
service httpd restart
```

### 5.1.5 Instalación de la Base de Datos MySQL.

Para instalar la última versión del servidor mysql y la librería de php para trabajar con mysql, se debe ejecutar el siguiente comando en la consola:

```
Yum install mysql-server php-mysql
```

Luego para iniciar el servidor mysql se debe ejecutando el siguiente comando:

```
service mysql Start
```

El archivo de configuración de mysql se encuentra en /etc/my.cnf.

Una vez instalado el servidor Apache y MySQL se debe configurar el sistema para que cada vez que se inicie el PC éstos servicios se inicien de igual forma, para ello, se debe acceder al menú Sistema/Administración/Servicio, que se encuentra en el menú del sistema, una vez dentro del panel de Configuración de Servicio, se debe seleccionar y marcar con un “*ticket*” los servicios httpd y mysqld, posteriormente apretar el botón Guardar.

## 5.1.6 Instalación de phpMyAdmin

PhpMyAdmin es un software de administración basado en Web que es fácil de configurar y de utilizar, además permite realizar casi todas las tareas necesarias sobre la base de datos MySQL.

Para instalar phpMyAdmin se debe ejecutar el siguiente comando en la consola:

```
Yum install phpMyAdmin
```

Luego de instalar phpMyAdmin hay que modificar el archivo de configuración de éste, para ello, se debe ejecutar en la consola el siguiente comando:

```
gedit /etc/phpMyAdmin/config.inc.php
```

Luego se buscan las siguientes líneas de código:

```
$cfg['Servers'][$i]['auth_type'] = 'http';  
$cfg['Servers'][$i]['user'] = '';  
$cfg['Servers'][$i]['password'] = '';
```

Y se reemplazan por las siguientes líneas de código:

```
$cfg['Servers'][$i]['auth_type'] = 'cookie';  
$cfg['Servers'][$i]['user'] = 'root';  
$cfg['Servers'][$i]['password'] = '';
```

En la línea donde debe estar el *password* del usuario root, se deja en blanco, ya que por defecto mysql crea un usuario *root* sin *password*, por lo tanto, se ingresará el *password* una vez que se ingrese a la base de datos MySQL por medio del administrador phpMyAdmin.

Luego se debe agregar la siguiente línea de código:

```
$cfg['blowfish_secret'] = 'una pregunta secreta'; //esto es para poner una pregunta  
secreta o clave
```

Y posteriormente se debe reiniciar el Servidor Apache:

```
service httpd restart
```

Una vez que se hayan realizados todos los pasos anteriores, se debe ingresar en el navegador web la dirección: <http://localhost/phpMyAdmin>

Se desplegará una pantalla donde se debe ingresar el Nombre de Usuario (*root*) y *Password* (en blanco).

Una vez dentro del Administrador phpMyAdmin hay que ingresar en el Item Privilegios, y modificar el usuario *root*, ingresando un *Password* para mayor seguridad.

### 5.1.7 Instalación de J2SDK

Para instalar J2SDK, se debe ejecutar el siguiente comando en la consola:

```
rpm -Uvh j2sdk-1_4_2_15-linux-i586.rpm
```

J2sdk quedará instalado en la carpeta `/usr/java/j2sdk1.4.2_15`.

### 5.1.8 Instalación de Libpng

Libpng es la librería de referencia oficial del formato de imágenes PNG<sup>36</sup>. Es una librería independiente de plataforma que contiene funciones en C para manejar imágenes que utilicen el formato PNG.

Para instalar esta librería se deben ejecutar los siguientes comandos:

```
tar -zxvf libpng-1.2.19.tar.gz
cd libpng-1.2.19
./configure
make
make test
make install
```

### 5.1.9 Instalación de Libjpeg

Libjpeg es una librería que permite manejar imágenes que utilicen el formato JPEG<sup>37</sup>.

---

<sup>36</sup> PNG son siglas en inglés de *Portable Network Graphics* (Formato gráfico basado en algoritmo de compresión sin pérdida para bitmaps).

Para instalar esta librería se deben ejecutar los siguientes comandos:

```
tar zxvf jpegsrc.v6b.tar.gz
cd jpeg-6b
./configure
make
make install
```

### 5.1.10 Instalación de la Librería GD

GD es una librería de gráficos para php, para instalar esta librería se deben ejecutar los siguientes comandos.

```
Yum install php-gd
```

### 5.1.11 Instalación de la Librería CURL

CURL es una herramienta usada por el intérprete de comandos, para transferir archivos con sintaxis URL.

Para instalar esta librería se deben ejecutar los siguientes comandos:

```
tar zxvf curl-7.16.4.tar.gz
cd curl-7.16.4
./configure
make
make install
```

### 5.1.12 Instalación de la Librería Tidy

Tidy es una herramienta que corrige, valida y diagnostica la validez de cualquier documento HTML. Al utilizar php-tidy permite a php 5 tomar ventaja directa de esta librería.

Para instalar esta librería se deben ejecutar los siguientes comandos:

```
Yum install php-tidy
```

---

<sup>37</sup>JPEG siglas en inglés de *Joint Photographic Experts Group* (Algoritmo diseñado para comprimir imágenes con 24 bits de profundidad o en escala de grises).

### 5.1.13 Instalación de ImageMagick

ImageMagick es una aplicación que sirve para crear, editar y componer imágenes. Puede leer, convertir y guardar imágenes en una gran variedad de formatos.

Para instalar ImageMagick se deben ejecutar los siguientes comandos:

```
tar zxvf ImageMagick-6.25-4.tar.gz
cd ImageMagick-6.25
./configure
make
make install
```

### 5.1.14 Instalación de Graphviz

Graphviz es un conjunto de herramientas que generan representaciones gráficas de grafos, organizando los nodos conforme a distintos criterios.

Para instalar Graphviz se deben ejecutar los siguientes comandos:

```
tar zxvf graphviz-2.12-1.src.rpm
cd graphviz-2.12
./configure
make
make install
```

*Nota: No descomprimir el archivo en el escritorio del usuario.*

Una forma de verificar si la instalación fue realizada correctamente es verificar en la carpeta /usr/local/bin que se encuentre el archivo “dot”.

### 5.1.15 Instalación de JHOVE

JHOVE es un programa que verifica de forma automática el formato y estado de conservación de los ficheros.

Para instalar JHOVE se deben ejecutar los siguientes comandos:

```
tar zxvf jhove-1_1f.tar.gz
mv jhove /usr/local
chown -R root:root jhove
cd /usr/local/jhove
./configure.pl /usr/local/jhove /usr/java /usr/java/
j2sdk1.4.2_15/jre
```

Luego se debe dar permiso de ejecución al archivo `/usr/local/jhove/jhove` ejecutando los siguientes comandos:

```
chmod 755 jhove
```

Por último se debe editar el archivo `jhove` ejecutando los siguientes comandos:

```
gedit /usr/local/jhove/jhove
```

Luego buscar las siguientes líneas de código:

```
JAVA_HOME=/usr/java  
JAVA=/usr/java/  
${JAVA} -classpath $CP Jhove $ARGS
```

Y reemplazarlas por las siguientes líneas de código:

```
JAVA_HOME=/usr/java/j2sdk1.4.2_15  
JAVA=$JAVA_HOME/bin/java  
${JAVA} -classpath $CP Jhove -c ${JHOVE_HOME}/conf/jhove.conf $ARGS
```

### 5.1.16 Instalación de domxml

Domxml son un conjunto de funciones que permiten operar sobre documentos XML con la API DOM.

Para instalar domxml se deben ejecutar los siguientes comandos:

```
Yum install php-domxml
```

### 5.1.17 Instalación de FEDORA-2.1.1

Antes de instalar Fedora se debe configurar el archivo *profile* que se encuentra en `/etc/`

Para ello, se debe ejecutar los siguientes comandos:

```
cd /etc
```

```
gedit profile
```

Luego buscar las siguientes líneas de código en el archivo *profile*

```
HISTSIZE=1000

If [-z "$INPUTRC" -a ! -f "$HOME/.inputrc" ]; ....
```

Entre las 2 líneas de código encontradas anteriormente se deben escribir las siguientes líneas de código:

```
JAVA_HOME=/usr/java/j2sdk1.4.2_15
FEDORA_HOME=/usr/local/Fedora-2.1.1
PATH=/usr/bin:/usr/sbin:/bin:/sbin:/usr/local/bin:/usr/local/sbin:/JAV
A_HOME/bin:$FEDORA_HOME/server/bin:/usr/mysql/bin
```

Una vez configurado el archivo *profile* se deben exportar las variables de entorno ejecutando los siguientes comandos:

```
export FEDORA_HOME=/usr/local/Fedora-2.1.1
export JAVA_HOME=/usr/java/j2sdk1.4.2_15
```

Luego se descomprimirá el archivo en donde se encuentra Fedora-2.1.1, y se dejará en la carpeta */usr/local*. Posteriormente se creará la base de datos de Fedora-2.1.1 que se llamará "Fedora21", con nombre de usuario y contraseña "FedoraAdmin", para ello, se debe ejecutar los siguientes comandos:

```
mv Fedora-2.1.1-server.tar.gz /usr/local
cd /usr/local
tar zxvf Fedora-2.1.1-server.tar.gz
chown -R root:root Fedora-2.1.1
cd /usr/local/Fedora-2.1.1/server/bin
mysql-config /usr root "(contraseña root)" FedoraAdmin FedoraAdmin Fedora21
mysql41_flag
```

Una vez creada la base de datos de Fedora-2.1.1, se debe ejecutar el archivo Fedora-setup, indicando la configuración de seguridad del repositorio digital Fedora-2.1.1, la cuál será “no-ssl-authenticate-apim”.

```
cd /usr/local/Fedora-2.1.1/server/bin
Fedora-setup no-ssl-authenticate-apim
```

Posteriormente se debe editar el archivo de configuración del servidor Fedora (Fedora.fcfg), el cual se encuentra en /usr/local/Fedora-2.1.1/server/config

- 1- Editar el Nombre del Repositorio Digital (repositoryName)
- 2- Editar la Dirección de email (adminEmailList)
- 3- Editar el PID Namespace (pidNamespace), aquí se debe ingresar el nombre base, con el cuál, se generará el identificador único de los nuevos objetos que serán ingresados al Repositorio Digital Fedora.
- 4- Editar el valor del “retainPIDs” que viene por defecto, ingresando el mismo valor del “pidNamespace” ingresado anteriormente.
- 5- Editar el Nombre del Repositorio Digital para OAI (repositoryName)

El último paso antes de correr el repositorio digital Fedora-2.1.1 es editar los archivos “Fedora-start” y “Fedora-stop”, que se encuentran en /usr/local/Fedora-2.1.1/server/bin. Para esto, se debe ejecutar los siguientes comandos:

```
gedit /usr/local/server/bin/Fedora-start
```

Luego ingresar las siguientes líneas de código al comienzo del archivo, después de las primeras líneas de comentarios:

```
FEDORA_HOME=/usr/local/Fedora-2.1.1
JAVA_HOME=/usr/java/j2sdk1.4.2_15
export FEDORA_HOME JAVA_HOME
```

Posteriormente se repite lo mismo para el archivo Fedora-stop.

Finalmente se debe ejecutar el siguiente comando para hacer partir al repositorio digital Fedora-2.1.1

```
cd /usr/local/Fedora-2.1.1/server/bin
Fedora-start
```

### 5.1.18 Instalación del front-end Fez 1.3

Lo primero que se debe hacer es configurar el SELinux, para esto se debe realizar los siguientes pasos:

```
-Ir a Sistema->Administración ->SELinux Management
-Ingresar la contraseña del usuario root
-Seleccionar Status
-Luego configurar los parámetros de la siguiente forma:
  System Default Enforcing Mode-> Permissive
  Current Enforcing Mode ->Permissive
  System Default Policy Type->targeted
```

Luego se debe editar las siguientes líneas de código del archivo php.ini que se encuentra en /etc.

```
gedit /etc/php.ini
Las siguientes líneas de código del archivo php.ini deben quedar de la siguiente forma:
allow_call_time_pass_reference= On
upload_max_filesize=100M
memory_limit= 256M
post_max_size=120M
```

Una vez editado el archivo php.ini se debe descomprimir el archivo .tar.gz y dejarlo en la carpeta /var/www/html, para ello, se debe ejecutar los siguientes comandos:

```
cp Fez-1.3.tar.gz /var/www/html/
cd /var/www/html/
tar zxvf Fez-1.3.tar.gz
mv Fez-1.3 Fez
chown -R apache:apache Fez
```

Luego se debe configurar el servidor Apache, editando el archivo /etc/httpd/conf/httpd.conf, en el cuál, se deben editar las siguientes líneas de código:

```
Agregar estas 2 líneas:
AddType application/x-httpd-php .php .phtml
AddType application/x-httpd-php-source .phps
Editar la siguiente línea:
```

```
DirectoryIndex index.php index.html index.html.var
```

Posteriormente se debe reiniciar el servidor Apache

```
service httpd restart
```

Para instalar Fez 1.3 se debe escribir la siguiente dirección en el navegador web:

<http://localhost/Fez/setup>

Luego se debe rellenar el formulario de la siguiente manera:

**Your Organisation:** Nombre de la Organización  
**Your Organisation (Initials):** Iniciales de la Organización  
**Application Name:** Nombre de la Aplicación  
**Application Administrator Email Address:** Mail del Administrador  
**Server Hostname:** localhost  
**Fez Relative URL:** /Fez/  
**Installation Path:** /var/www/html/Fez/  
**Fedora Server Setup:** Seleccionar no-ssl-authenticate-apim  
**Fedora Version:** 2.1.1  
**Fedora Server Base Location:** https://localhost:8080/Fedora  
**Fedora Username:** FedoraAdmin  
**Fedora Password:** FedoraAdmin  
**Fedora PID Namespace :** Valor del PID Namespace que se ingreso anteriormente en el archivo Fedora.fcfg  
**Fedora Database Type:** mysql  
**Fedora Database Server Hostname:** localhost:8080/Fedora  
**Fedora Database Server Port:** 3306  
**Fedora Database Name:** Fedora21  
**Fedora Database Username:** FedoraAdmin  
**Fedora Database Password:** FedoraAdmin  
**Fez Database Server Hostname:** localhost.localdomain  
**Fez Database Name:** Fez (Seleccionar con un tick “Create Database” y “Drop Tables If They Already Exist”)  
**Fez Database Username:** root  
**Fez Database Password:** Contraseña del usuario root de mysql

Una vez ingresado los datos al formulario de instalación de Fez, se debe apretar el botón “Start Installation” para instalar Fez con su correspondiente Base de datos.

El siguiente paso es configurar el archivo config.inc.php que se encuentra en /var/www/html/Fez/config.inc.php, editando las siguientes líneas de código que deben quedar de la siguiente forma:

```
@define("APP_DOT_EXEC", "/usr/local/bin/dot");
@define("APP_PHP_EXEC", "/usr/bin/php");
@define("APP_PDFTOTEXT_EXEC", "/usr/local/bin/pdftotext");

@define("APP_FEDORA_LOCATION", "localhost:8080/Fedora");

@define("WEBSERVER_LOG_DIR", "/etc/httpd/logs/");

@define("APP_DEFAULT_USER_TIMEZONE", "America/Santiago");
```

Para verificar si se instaló todo correctamente se debe escribir la siguiente dirección en el navegador web: [http://localhost/Fez/setup/check\\_sanity.php](http://localhost/Fez/setup/check_sanity.php)

Finalmente para Ingresar al Repositorio Digital se debe ingresar la siguiente dirección en el navegador web: <http://localhost/Fez>

## 5.2 Creación del Objeto Digital Utilizando el Estándar de Metadatos MODS.

Se implementará el repositorio digital para la Universidad Austral de Chile, para que funcione como un sistema de administración de documentos de tesis, por lo tanto, el tipo de documentos que se ingresarán al sistema serán Tesis.

Para la creación del objeto digital que representará el documento de tesis, se deberá editar los metadatos (objetos XSD) que están predefinidos en el sistema, y que interactúan entre sí, los cuáles son los siguientes:

1. **MODS**: Es el cuál define el estándar del contenido del documento de tesis.
2. **RELS-EXT**: Es el objeto XSD que asocia el documento de tesis con una determinada colección.
3. **FezMD**: Es el objeto XSD que define los datos para el sistema Fez, como fecha de creación, fecha de actualización, además de registrar el ID de cada registro (Documento de Tesis) y el nombre del usuario que lo ingreso.
4. **Fedora Object XML**: Es el principal objeto XSD, ya que él interactúa con el sistema y se relaciona con los demás metadatos.

Para poder editar los objetos XSD mencionados anteriormente, se debe tener permiso de Administrador, y se debe ingresar en Administración -> *Manage Document Type XSDs*. Como se muestra en la figura 22.



Figura 22: Acceso a *Manage Document Type XSDs*

Una vez que se haya ingresado en *Manage Document Type XSDs*, se mostrará a la derecha de la pantalla una tabla llamada “*Existing Document Type XSDs*” dónde mostrará todos los objetos XSD creados.

### 5.2.1 Editar MODS

Para editar el objeto MODS se deberá realizar los siguientes pasos:

1. Desplegar los objetos del tipo MODS, para ésto se deberá hacer “*click*” en *Edit* en la columna *XSD Display* de la fila MODS, como se muestra en la figura 23.

Existing Document Type XSDs:						
All	Title	Version	Top Element Name	Element Prefix	XSD Source	XSD Displays
<input type="checkbox"/>	AudioMD	1.0	AUDIOMD		Edit	Edit
<input type="checkbox"/>	BookMD	1.0	BookMD		Edit	Edit
<input type="checkbox"/>	ChapterMD	1.0	ChapterMD		Edit	Edit
<input type="checkbox"/>	ConferencePaperMD	1.0	ConferencePaperMD		Edit	Edit
<input type="checkbox"/>	ConferencePosterMD	1.0	ConferencePosterMD		Edit	Edit
<input type="checkbox"/>	ConferenceProceedingsMD	1.0	ConferenceProceedingsMD		Edit	Edit
<input type="checkbox"/>	DeptTechReportMD	1.0	DeptTechReportMD		Edit	Edit
<input type="checkbox"/>	DesignMD	1.0	DesignMD		Edit	Edit
<input type="checkbox"/>	Fedora Object XML	1.1	digitalObject	foxml	Edit	Edit
<input type="checkbox"/>	FezACML	1.0	FezACML		Edit	Edit
<input type="checkbox"/>	FezHistory	1.0	FezHistory		Edit	Edit
<input type="checkbox"/>	FezMD	1.0	FezMD		Edit	Edit
<input type="checkbox"/>	geospatialMD	1.0	geospatialCoveragesMD		Edit	Edit
<input type="checkbox"/>	JournalArticleMD	1.0	JournalArticleMD		Edit	Edit
<input type="checkbox"/>	MagazineArticleMD	1.0	MagazineArticleMD		Edit	Edit
<input type="checkbox"/>	MARCXML	1.1	collection		Edit	Edit
<input type="checkbox"/>	MODS	3.2 without Related	mods	mods	Edit	Edit

Figura 23: Despliegue de los objetos XSD MODS

2. Copiar el objeto “*Thesis MODS*”, al cuál se le deberá cambiar el nombre “*Thesis MODS*” por un nombre relacionado con el documento que será contenido en el repositorio digital (ej: Tesis estándar Uach MODS), para ello, se deberá hacer

“click” en “Clone” en la columna “Clone Display”, como se muestra en la figura 24.

Existing XSD Displays:						
All	Title	Version	Object Type	Enabled?	XSD HTML Matching Editor	Clone Display
<input type="checkbox"/>	Newspaper Article MODS	1.0	(4) Reference	yes	Edit	Clone
<input type="checkbox"/>	Online Journal Article MODS	1.0	(4) Reference	yes	Edit	Clone
<input type="checkbox"/>	Patent MODS	1.0	(4) Reference	yes	Edit	Clone
<input type="checkbox"/>	Patent RQF 2007 MODS	1.0	(4) Reference	yes	Edit	Clone
<input type="checkbox"/>	Preprint MODS	1.0	(4) Reference	yes	Edit	Clone
<input checked="" type="checkbox"/>	Thesis MODS	1.0	(4) Reference	yes	Edit	Clone

Figura 24: Copia de objeto *Thesis* MODS.

3. Editar el objeto recientemente clonado haciendo “click” en “Edit”, en la columna “XSD HTML Matching Editor”, como se muestra en la figura 25:

Existing XSD Displays:						
All	Title	Version	Object Type	Enabled?	XSD HTML Matching Editor	Clone Display
<input type="checkbox"/>	Newspaper Article MODS	1.0	(4) Reference	yes	Edit	Clone
<input type="checkbox"/>	Online Journal Article MODS	1.0	(4) Reference	yes	Edit	Clone
<input type="checkbox"/>	Patent MODS	1.0	(4) Reference	yes	Edit	Clone
<input type="checkbox"/>	Patent RQF 2007 MODS	1.0	(4) Reference	yes	Edit	Clone
<input type="checkbox"/>	Preprint MODS	1.0	(4) Reference	yes	Edit	Clone
<input checked="" type="checkbox"/>	Tesis estandar Uach MODS	1.0	(4) Reference	yes	Edit	Clone
<input type="checkbox"/>	Thesis MODS	1.0	(4) Reference	yes	Edit	Clone
<input type="checkbox"/>	Working Paper MODS	1.0	(4) Reference	yes	Edit	Clone

Figura 25: Editando el objeto *Thesis* estándar Uach MODS.

4. Una vez dentro del editor de objetos XSD (XSD HTML Matching Editor), a la izquierda de la pantalla estarán los elementos xml que conforman el objeto XSD (Tesis estándar Uach MODS) y al hacer “click” en elemento xml se mostrará, a la derecha de la pantalla, los campos de configuración de cada elemento xml. Los principales campos de configuración que se deben editar en cada elemento xml de un objeto XSD son los siguientes:

- **Title:** Este campo es donde se ingresará el título del elemento xml.
- **Display Order Priority:** Este campo es dónde se ingresará el número que representará la prioridad del orden de despliegue del elemento xml en el form.

- **Enabled in Form?:** “Check Box” dónde se especificará si el elemento xml se mostrará en el *Form*.
- **Indexed?:** “Check Box” debe ser “tickeado” cuándo el elemento xml será usado para búsquedas en el sistema Fez (ej: Patrocinante).
- **Search Key:** “Combo Box” dónde se seleccionará el índice que representará al elemento xml para las búsquedas en el sistema.
- **Required?:** “Check Box” dónde se especificará que elemento xml será obligatorio en el *form*.
- **Multiple?:** “Check Box” dónde se especificará que serán ingresados múltiples datos en el elemento xml (ej: Autores, patocinantes).
- **Multiple Limit:** Este campo es dónde se especificará el número máximo de datos ingresados en el elemento xml.
- **Field Type:** Este campo es dónde se especificará que tipo de elemento xml se representará en el form (*Text Input, Text Area, Combo box, etc*).

Los elementos xml, los cuáles representarán los campos en el formulario, que hay que editar son los siguientes:

- **titleInfo->title:** El campo que representará éste elemento xml en el “Form” es del tipo “Text Input”, en el cuál, se ingresará el Título del documento de tesis.
- **name->namePart:** En éste elemento xml se crean 2 sub-elementos xml, dónde se ingresarán los autores y patrocinantes del documento de tesis, dónde cada sub-elemento xml representará en el “Form” un campo múltiple (Ingresar más de un Autor o Patrocinante) del tipo “Text Input”.
- **genre:** El campo que representará esté elemento xml en el “Form” es del tipo “Hidden Static Text”, en el cuál, se especificará el tipo de documento que se ingresará al repositorio digital.
- **originInfo->dateIssued:** El campo que representará esté elemento xml en el “Form” es del tipo “Date”, en el cuál, se ingresará la Fecha de Publicación del Documento de Tesis.
- **originInfo->copyrightDate:** El campo que representará esté elemento xml en el “Form” es del tipo “Text Input”, en el cuál, se ingresará el nombre que tendrá el Copyright del documento de tesis.

- **Language:** El campo que representará este elemento xml en el “Form” es del tipo “*Hidden Static Text*”, en el cuál, se ingresará el idioma del documento de tesis (Ej: es).
- **Abstract:** El campo que representará este elemento xml en el “Form” es del tipo “*Text Area*”, en el cuál, se ingresará el resumen del documento de tesis.
- **Note:** El campo que representará este elemento xml en el “Form” es del tipo “*Text Area*”, en el cuál, se ingresará el resumen del documento de tesis en idioma Ingles.
- **subject->Topic:** En este elemento xml se crearán 3 sub-elementos, dónde se ingresarán las palabras claves de búsqueda, la Facultad y la Escuela, dónde los sub-elementos representarán en el “Form” un campo del tipo “*Text Input*”, “*Controlled Vocabulary*”, “*Controlled Vocabulary*” correspondientemente.
- **subject-ID:** Cuando se crea un elemento xml del tipo “Control de Vocabulario” se debe crear un elemento xml del tipo “XSDMF ID Reference” que referencia al Control de Vocabulario.
- **relatedItem->name->namePart:** El campo que representará este elemento xml en el “Form” es del tipo “*Text Input*”, en el cuál, se ingresará el nombre de la Universidad a la cuál pertenece el documento de tesis.
- **relatedItem->genre:** El campo que representará este elemento xml en el “Form” es del tipo “*Controlled Vocabulary*”, en el cuál, se ingresará el Grado Obtenido por el Estudiante.
- **relatedItem->originInfo->Publisher:** El campo que representará este elemento xml en el “Form” es del tipo “*Hidden Static Text*”, en el cuál, indicará el Editor del Documento de Tesis (Universidad Austral de Chile).

Cuando se crea un sub-elemento en un elemento xml, se deberá crear un sub-elemento del tipo “*Hidden Static Text*”, el cuál estará relacionado con el sub-elemento creado para representar el campo en el *Form*. Esta relación se realiza en el elemento xml padre, donde se indica el elemento estático y el que representará al campo en el “Form”. Como se muestra en la figura 26.

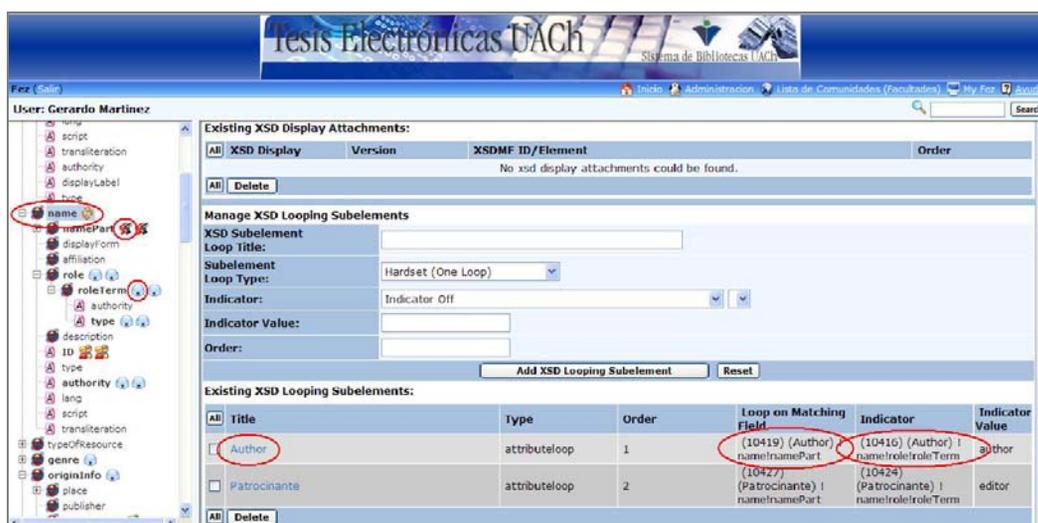


Figura 26: Relación del sub-elementos Autor.

### 5.2.2 Editar RELS-EXT

Para editar el objeto RELS-EXT se deberá realizar los siguientes pasos:

1. Desplegar los objetos del tipo RELS-EXT, para esto se deberá hacer “click” en “Edit” en la columna XSD Display de la fila RELS-EXT, como se muestra en la figura 27.

Existing Document Type XSDs:						
All	Title	Version	Top Element Name	Element Prefix	XSD Source	XSD Displays
<input type="checkbox"/>	AudioMD	1.0	AUDIOMD		Edit	Edit
<input type="checkbox"/>	BookMD	1.0	BookMD		Edit	Edit
<input type="checkbox"/>	ChapterMD	1.0	ChapterMD		Edit	Edit
<input type="checkbox"/>	ConferencePaperMD	1.0	ConferencePaperMD		Edit	Edit
<input type="checkbox"/>	ConferencePosterMD	1.0	ConferencePosterMD		Edit	Edit
<input type="checkbox"/>	ConferenceProceedingsMD	1.0	ConferenceProceedingsMD		Edit	Edit
<input type="checkbox"/>	RELS-EXT	1.0 - isMemberOf isD	RDF	rdf	Edit	Edit

Figura 27: Despliegue de los objetos XSD RELS-EXT.

2. Copiar el objeto “Fedora Record RELS-EXT Display ”, al cuál se le deberá cambiar el nombre por un nombre relacionado con el documento que será contenido en el Repositorio Digital (ej: Fedora Record Uach RELS-EXT Display), para ello, se deberá hacer “click” en “Clone” en la columna “Clone Display”, como se muestra en la figura 28.

Existing XSD Displays:

All	Title	Version	Object Type	Enabled?	XSD HTML Matching Editor	Clone Display
<input type="checkbox"/>	Fedora Collection RELS-EXT Display	1.0	(4) Reference	yes	Edit	Clone
<input type="checkbox"/>	Fedora Record RELS-EXT Display	1.0	(4) Reference	yes	Edit	Clone

All Delete

Figura 28: Copia de Objeto Fedora Record RELS-EXT Display.

3. Editar el objeto recientemente clonado haciendo “click” en “Edit” en la Columna “XSD HTML Matching Editor”, como se muestra en la figura 29:

Existing XSD Displays:

All	Title	Version	Object Type	Enabled?	XSD HTML Matching Editor	Clone Display
<input type="checkbox"/>	Fedora Collection RELS-EXT Display	1.0	(4) Reference	yes	Edit	Clone
<input type="checkbox"/>	Fedora Record Uach RELS-EXT Display	1.0	(4) Reference	yes	Edit	Clone
<input type="checkbox"/>	Fedora Record RELS-EXT Display	1.0	(4) Reference	yes	Edit	Clone

All Delete

Figura 29: Editando el objeto Fedora Record Uach RELS-EXT Display.

4. Una vez dentro del editor de objetos XSD (XSD HTML Matching Editor), a la izquierda de la pantalla estarán los elementos xml que conforman el Objeto XSD (Fedora Record Uach RELS-EXT Display). El Elemento xml que se deberá editar es el siguiente:

- **Description->isDerivationOf->resource:** Este elemento se deberá deshabilitar, ya que su función es relacionar 2 registros en forma descendente, y como el documento de tesis es uno solo por alumno se deberá deshabilitar esté elemento xml, para ello, se deberá hacer “click” sobre el elemento y sacar el ticket del campo “Enabled in form”, como se muestra en la figura 30.

Figura 30: Deshabilitar el elemento xml resource.

### 5.2.3 Editar FezMD

Para editar el objeto FezMD se deberá realizar los siguientes pasos:

1. Desplegar los objetos del tipo FezMD, para esto se deberá hacer “click” en “Edit” en la columna *XSD Display* de la fila FezMD, como se muestra en la figura 31.

Existing Document Type XSDs:						
All	Title	Version	Top Element Name	Element Prefix	XSD Source	XSD Displays
<input type="checkbox"/>	AudioMD	1.0	AUDIOMD		Edit	Edit
<input type="checkbox"/>	BookMD	1.0	BookMD		Edit	Edit
<input type="checkbox"/>	ChapterMD	1.0	ChapterMD		Edit	Edit
<input type="checkbox"/>	ConferencePaperMD	1.0	ConferencePaperMD		Edit	Edit
<input type="checkbox"/>	ConferencePosterMD	1.0	ConferencePosterMD		Edit	Edit
<input type="checkbox"/>	ConferenceProceedingsMD	1.0	ConferenceProceedingsMD		Edit	Edit
<input type="checkbox"/>	DeptTechReportMD	1.0	DeptTechReportMD		Edit	Edit
<input type="checkbox"/>	DesignMD	1.0	DesignMD		Edit	Edit
<input type="checkbox"/>	Fedora Object XML	1.1	digitalObject	foxml	Edit	Edit
<input type="checkbox"/>	FezACML	1.0	FezACML		Edit	Edit
<input type="checkbox"/>	FezHistory	1.0	FezHistory		Edit	Edit
<input type="checkbox"/>	FezMD	1.0	FezMD		Edit	Edit
<input type="checkbox"/>	geospatialMD	1.0	geospatialCoveragesMD		Edit	Edit

Figura 31: Despliegue de los objetos XSD FezMD.

2. Copiar el objeto “*Simple FezMD Display*”, al cuál se le deberá cambiar el nombre por un nombre relacionado con el documento que será contenido en el Repositorio Digital (ej: *Simple FezMD Uach Display*), para ello, se deberá hacer “click” en “Clone” en la columna “Clone Display”, como se muestra en la figura 32.

Existing XSD Displays:						
All	Title	Version	Object Type	Enabled?	XSD HTML Matching Editor	Clone Display
<input type="checkbox"/>	FezMD Collection Display	1.0	(4) Reference	yes	Edit	Clone
<input type="checkbox"/>	FezMD Community Display	1.0	(4) Reference	yes	Edit	Clone
<input type="checkbox"/>	FezMD Display	1.0	(4) Reference	yes	Edit	Clone
<input type="checkbox"/>	FezMD RQF Display	1.0	(4) Reference	yes	Edit	Clone
<input type="checkbox"/>	Person FezMD Display	1.0	(4) Reference	yes	Edit	Clone
<input type="checkbox"/>	Simple FezMD Display	1.0	(4) Reference	yes	Edit	Clone

Figura 32: Copia de objeto Simple FezMD Display.

3. Editar el objeto recientemente clonado haciendo “click” en “Edit” en la Columna “XSD HTML Matching Editor”, como se muestra en la figura 33.

Existing XSD Displays:						
All	Title	Version	Object Type	Enabled?	XSD HTML Matching Editor	Clone Display
<input type="checkbox"/>	FezMD Collection Display	1.0	(4) Reference	yes	Edit	Clone
<input type="checkbox"/>	FezMD Community Display	1.0	(4) Reference	yes	Edit	Clone
<input type="checkbox"/>	FezMD Display	1.0	(4) Reference	yes	Edit	Clone
<input type="checkbox"/>	FezMD RQF Display	1.0	(4) Reference	yes	Edit	Clone
<input type="checkbox"/>	Person FezMD Display	1.0	(4) Reference	yes	Edit	Clone
<input type="checkbox"/>	Simple FezMD Display	1.0	(4) Reference	yes	Edit	Clone
<input type="checkbox"/>	Simple FezMD Uach Display	1.0	(4) Reference	yes	Edit	Clone

Figura 33: Editando el objeto Simple FezMD Uach Display.

4. Una vez dentro del editor de objetos XSD (XSD HTML *Matching Editor*), a la izquierda de la pantalla estarán los elementos xml que conforman el objeto XSD (*Simple FezMD Uach Display*). Los Elemento xml que se deberá editar son los siguiente:

- **Refereed:** Este elemento xml se deberá deshabilitar ya que el “Form” del documento de tesis no tendrá un campo de referencia, para ello se deberá hacer “click” sobre el elemento y sacar el ticket del campo “Enabled in form”.
- **Reference\_text:** Este elemento xml se deberá deshabilitar ya que el form del documento de tesis no tendrá un campo de referencia, para ello se deberá hacer “click” sobre el elemento y deseleccionar el campo “Enabled in form”, como se muestra en la figura 34.

The screenshot shows the XSD HTML Matching Editor interface. On the left, a tree view displays XML elements for 'Simple FezMD Uach Display'. The elements 'refereed' and 'reference' are circled in red. On the right, the configuration panel for the selected element is shown. The 'Enabled in Form?' checkbox is circled in red, and the 'reference' element is marked as 'DISABLED Check Box'.

Figura 34: Deshabilitar el elemento xml *refereed* y *referente\_text*.

## 5.2.4 Editar Fedora Object XML

Para editar el objeto Fedora Object XML se deberá realizar los siguientes pasos:

1. Desplegar los objetos del tipo Fedora Object XML, para esto se deberá hacer “click” en “Edit” en la columna *XSD Display* de la fila *Fedora Object XML*, como se muestra en la figura 35:

Existing Document Type XSDs:						
All	Title	Version	Top Element Name	Element Prefix	XSD Source	XSD Displays
<input type="checkbox"/>	AudioMD	1.0	AUDIOMD		Edit	Edit
<input type="checkbox"/>	BookMD	1.0	BookMD		Edit	Edit
<input type="checkbox"/>	ChapterMD	1.0	ChapterMD		Edit	Edit
<input type="checkbox"/>	ConferencePaperMD	1.0	ConferencePaperMD		Edit	Edit
<input type="checkbox"/>	ConferencePosterMD	1.0	ConferencePosterMD		Edit	Edit
<input type="checkbox"/>	ConferenceProceedingsMD	1.0	ConferenceProceedingsMD		Edit	Edit
<input type="checkbox"/>	DeptTechReportMD	1.0	DeptTechReportMD		Edit	Edit
<input type="checkbox"/>	DesignMD	1.0	DesignMD		Edit	Edit
<input type="checkbox"/>	Fedora Object XML	1.1	digitalObject	foxml	Edit	Edit
<input type="checkbox"/>	FezACML	1.0	FezACML		Edit	Edit
<input type="checkbox"/>	FezHistory	1.0	FezHistory		Edit	Edit
<input type="checkbox"/>	FezMD	1.0	FezMD		Edit	Edit
<input type="checkbox"/>	geospatialMD	1.0	geospatialCoveragesMD		Edit	Edit
<input type="checkbox"/>	JournalArticleMD	1.0	JournalArticleMD		Edit	Edit
<input type="checkbox"/>	MagazineArticleMD	1.0	MagazineArticleMD		Edit	Edit
<input type="checkbox"/>	MARCXML	1.1	collection		Edit	Edit

Figura 35: Despliegue de los objetos XSD Fedora *Object XML*.

2. Copiar el objeto “*Generic Document*” versión MODS, al cuál se le deberá cambiar el nombre “*Generic Document*” por el nombre del documento que será contenido en el repositorio digital (ej: Tesis Uach), para ello, se deberá hacer “click” en “Clone” en la columna “*Clone Display*”, como se muestra en la figura 36.

All	Title	Version	Object Type	Enabled?	XSD HTML Matching Editor	Clone Display
<input type="checkbox"/>	Digilib Image	1.0	(3) Record	yes	Edit	Clone
<input type="checkbox"/>	Fedora Collection Display	1.0	(2) Collection	yes	Edit	Clone
<input type="checkbox"/>	Fedora Community Display	1.0	(1) Community	yes	Edit	Clone
<input type="checkbox"/>	Fedora RQF 2006 Collection Display	Dublin Core 1.0	(2) Collection	yes	Edit	Clone
<input type="checkbox"/>	Generic Document	1.0	(3) Record	no	Edit	Clone
<input type="checkbox"/>	Generic Document	MODS 1.0	(3) Record	yes	Edit	Clone

Figura 36: Copia de Objeto Generic Document.

3. Editar el objeto recientemente clonado haciendo “click” en “Edit” en la Columna “*XSD HTML Matching Editor*”, como se muestra en la figura 37.

All	Title	Version	Object Type	Enabled?	XSD HTML Matching Editor	Clone Display
<input type="checkbox"/>	Software	1.0	(3) Record	no	Edit	Clone
<input checked="" type="checkbox"/>	Tesis Uach	MODS 1.0	(3) Record	yes	Edit	Clone

Figura 37: Editando el objeto Tesis Uach.

4. Una vez dentro del editor de objetos XSD (XSD HTML *Matching Editor*), a la izquierda de la pantalla estarán los elementos xml que conforman el objeto XSD (Tesis Uach). Los elementos xml que hay que editar son los siguientes:

- ***objectProperties->property->Value***: Se deberá hacer “click” en XSDMF ID Reference, este elemento xml hace referencia al objeto XSD que representará el estándar del contenido del documento de tesis (Estándar MODS), por lo tanto, en la parte derecha de la pantalla en el campo XSDMF ID Reference se deberá seleccionar el objeto “Tesis estándar Uach MODS Version 1.0”, que anteriormente fue creado, como se muestra en la figura 38.

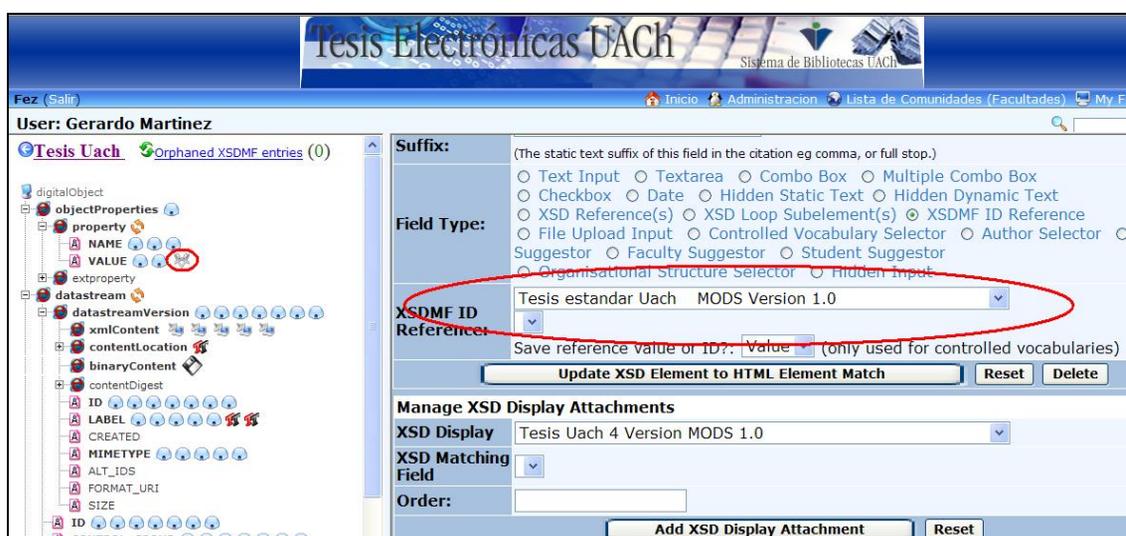


Figura 38: Editando el elemento xml *objectProperties -> property->Value*

- ***datastream->datastreamVersion->xmlContent***: Es donde se crearán las referencia hacia todos los objetos XSD que contribuyen para la correcta representación del contenido que se desea registrar (Documento de Tesis). Este campo tiene 5 sub-elementos los cuáles son: DC, FezMD, MODS, RELS-EXT, FezACML, donde estos mediante el campo “XSD Display Reference” hacen referencia a los objetos *Generic MODS DC Display, Simple FezMD Display, Generic MODS, Fedora Record RELS-EXT Display, FezACML for Records*, correspondientemente. FezACML

no se editará ya que este objeto administra los roles de seguridad del sistema, por lo tanto, los sub-elementos que se editarán son MODS, RELS-EXT y FezMD, cambiando sus referencias por las creadas anteriormente (Tesis estándar Uach MODS, Fedora Record Uach RELS-EXT Display, Simple FezMD Uach Display). Como se muestra en la figura 39.

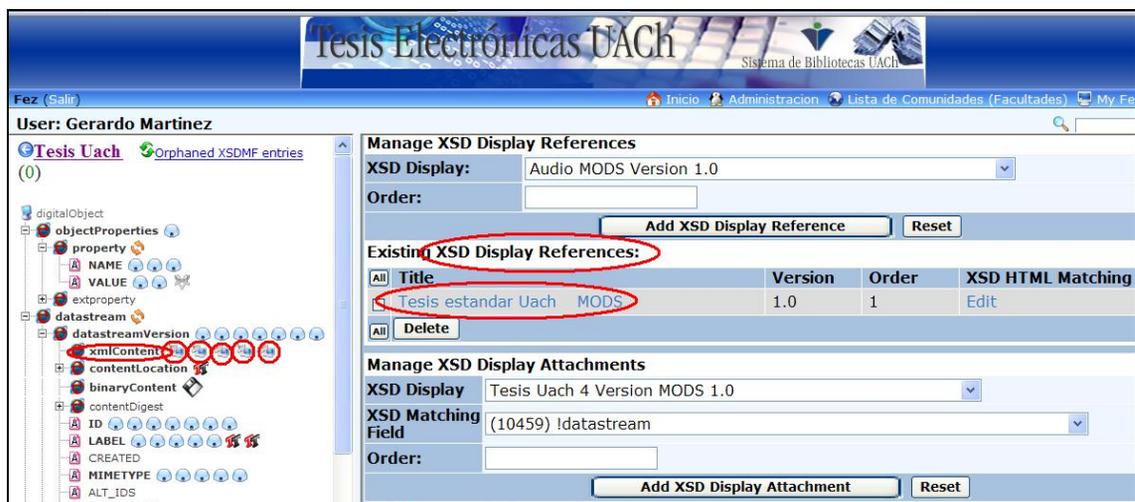


Figura 39: Editando el elemento xml *datastream*->*datastreamVersion*->*xmlContent*

### 5.3 Creación de Índices de Búsquedas

Para la creación de los índices de búsqueda se deberá ingresar en Administrador->*Manage Search Keys* como lo muestra la figura 40.



Figura 40: Ingresar al Administrador->*Manage Search Keys*.

Luego se deberá ingresar los siguientes datos:

- **Título:** Nombre del índice de Búsqueda.
- **Alternative Title:** Nombre alternativo del índice, este campo es utilizado para cambiar el nombre de los índices predefinidos y traducirlos al español sin que

cambie el funcionamiento interno del sistema, ya que este nombre se mostrará al usuario pero el sistema trabajará con el “Title”.

- **Use in Simple Search?:** Este Campo debe ser “tickeado” si el índice será usado en las búsquedas simples del sistema.
- **Show in Advanced Search Form?:** Este Campo es debe ser “tickeado” para que se muestre como opción de búsqueda en la búsqueda avanzada del sistema.
- **Show in My Fez Search Form?:** Este Campo debe ser “tickeado” para que se muestre como opción de búsqueda en My Fez del sistema.
- **Display Order:** Numero de orden de despliegue en el sistema.
- **Field Type:** En este campo se deberá seleccionar el tipo del campo que será indexado al índice.

Como se muestra en la figura 41.

Manage Search Keys	
Title:	<input type="text"/>
Alternative Title (if empty Title will be used, useful for language translations of Fez):	<input type="text"/>
Use in Simple Search?:	<input type="checkbox"/> <small>(If this tickbox is enabled any fields using this search key will be included in a simple search)</small>
Show in Advanced Search Form?:	<input type="checkbox"/> <small>(If this tickbox is enabled this search key will show up as an option checkbox in the advanced search form)</small>
Show in My Fez Search Form?:	<input type="checkbox"/> <small>(If this tickbox is enabled this search key will show up as an option checkbox in the My Fez search form)</small>
Display Order:	<input type="text"/>
Multiple?:	<input type="checkbox"/> <small>(If this enabled tickbox is ticked this element will be able to be entered multiple times in forms)</small>
Multiple Limit:	<input type="text"/> <small>(The maximum amount of html input elements that will show for this element if it is a multiple)</small>
Field Type:	<input type="radio"/> Text Input <input type="radio"/> Text Area <input type="radio"/> Combo Box <input type="radio"/> Multiple Combo Box <input type="radio"/> Check Box <input type="radio"/> Date Selector <input type="radio"/> Controlled Vocabulary <input type="radio"/> All Controlled vocabularies
Function Lookup for Value of ID (optional):	<input type="text"/>
<input type="button" value="Add Search Key"/> <input type="button" value="Reset"/>	

Figura 41: Creación de un Índice de Búsqueda.

## 5.4 Creación de Comunidades y Colecciones

Para la creación de las comunidades se deberá ir a “Lista de Comunidades” y hacer “click” en “Crear Nueva Comunidad” como se muestra en la figura 42.



Figura 42: Ingresar a Lista de Comunidades-> Crear Nueva Comunidad.

Luego se deberá ingresar los siguientes datos:

- **Name:** Se deberá ingresar el nombre de la Comunidad.
- **XSD Display Document Types:** Se deberá seleccionar “*Fedora Collection Display Versión 1.0*”.
- **Description:** Se deberá ingresar la descripción de la comunidad.
- **Keyword(s):** Se deberá ingresar la(s) palabra(s) de búsqueda(s). Como se muestra es la figura 43.

Figura 43: Creación de una Nueva Comunidad.

Una vez creada la comunidad, al ingresar en ella se podrá crear las colecciones, las cuáles representarán las Escuelas de la Universidad y contendrán los documentos de tesis. Al hacer “click” en Crear se deberá rellenas los siguientes campos:

- **Title:** Se deberá ingresar el nombre de la colección (Escuela).
- **XSD Display Document Types:** Se deberá ingresar el tipo de documentos XSD que serán registrados en la colección, en esté caso será el objeto XSD que se creó anteriormente “Tesis Uach Versión MODS 1.0” (objeto digital) que registrará a los documentos de tesis.
- **Member of Communities:** Se deberá seleccionar la comunidad a la cual pertenecerá esta colección.

- **Description:** Se deberá ingresar la descripción de la colección.
- **Keyword:** Se deberán ingresar la(s) palabra(s) clave(s) de búsqueda.

Al finalizar hacer “click” en Publicar, como se muestra en la figura 44.

Figura 44: Creación de una Nueva Colección.

## 5.5 Administración de Políticas de Seguridad para las Comunidades y Colecciones.

Para la administración de los roles de seguridad o privilegios de acceso de las comunidades y colecciones, se deberá hacer “click” en el icono  (*Edit Security*) que se encuentra al lado del nombre de la comunidad o colección.

Una vez dentro del Editor de Seguridad (*Edit Security*) de la comunidad o colección, están los 5 principales roles de seguridad, los cuales son asignados a los usuario o grupos de usuarios. Los roles de seguridad son los siguientes:

1. **Lister:** Permite que los usuarios puedan ver el objeto (comunidad o colección) en los listados y resultados de búsqueda. Por defecto todos los usuarios pueden ver los objetos en los listados y resultados de búsqueda.
2. **Viewer:** Permite que los usuarios vean el objeto (comunidad o colección). Por defecto todos los usuarios pueden ver los objetos, además si a un usuario se le asigna este rol sobre la comunidad o colección, por defecto al usuario se le asignará el rol *Lister*.
3. **Creator:** Permite a los usuarios crear objetos (comunidades, colecciones o registros) sobre un objeto (comunidad o colección) determinado. Si un usuario tiene asignado el Rol Creator, por defecto se le asigna el rol *Viewer*.

4. **Editor:** Permite a los usuario poder editar un objeto (comunidad o colección) y a sus objetos hijos. Si un usuario tiene asignado el rol Editor, por defecto se le asigna el rol *Viewer*.
5. **Approver:** Permite a los usuarios poder publicar archivos de un determinado objeto (comunidad o colección), que están en espera de ser publicados.

En el repositorio digital las asignaciones de roles de seguridad son los siguientes:

- A los grupos de alumnos se les asignará el rol de seguridad “*Creator*” sobre solo la colección (Escuela) a la cuál pertenecen, ya que cada Escuela de la Universidad tendrá un grupo de usuarios. Ej: El grupo de usuarios de Ingeniería Civil en Informática solo tendrá el rol *Creator* (Para poder Ingresar Documentos de Tesis al sistema) sobre la Colección “Escuela de Ingeniería Civil en Informática”.
- Los grupos que no tiene el privilegio de administrador, al ingresar sus documentos de tesis al sistema, estos quedan en un *workflow* en espera ha ser aprobados, los cuáles son aprobados o desaprobados por el grupo de la biblioteca que tendrá el rol de *Approver*.

## **5.6 Creación de una API para la Conexión entre el Repositorio Digital y la Base de Datos de una Organización para la Autenticación de los Usuarios.**

Para mejorar el sistema se creó una API, donde todo lo concerniente con los usuarios del sistema (Autenticación, asignación de roles de seguridad,..), interactuase de forma remota (Mediante el protocolo XML RPC), entre el servidor del repositorio digital y el servidor de la organización. El motivo de esta creación fue para que al momento de implementar el sistema en una organización, el sistema interactúe con la base de datos que ya consta la organización, para no migrar los datos de los usuarios a la base de datos del repositorio digital. Para ello, se deberá editar las siguientes clases en el sistema:

- Class.user.php
- Class.group.php
- Class.auth.php ( Solo la Función “*userExists()*” )
- Class.prefs.php (Solo las Funciones “*set()*” y “*get()*” ).

Los procesos que realiza el sistema cuando un usuario se autentifica en el sistema son los siguientes:

- Usuario ingresa su nombre de usuario y contraseña en la pagina login.php

- Login.php llama a la clase user.php en el servidor del repositorio digital.
- La clase user.php le envía los datos ingresados por el usuario, mediante el protocolo xml-rpc, al servidor de la organización.
- El servidor de la organización recibe los datos del usuario y lo valida con su base de datos, luego envía los datos de respuesta al servidor del repositorio digital mediante el protocolo xml-rpc.
- Una vez que el servidor del repositorio digital recibe los datos del servidor de la organización, saca los datos del protocolo xml-rpc y con los datos necesarios de su propia base de datos, despliega la página de inicio del sistema (index.php) mostrando los contenidos del sistema dependiendo de que tipo de usuario es, como se muestra en la figura 45.

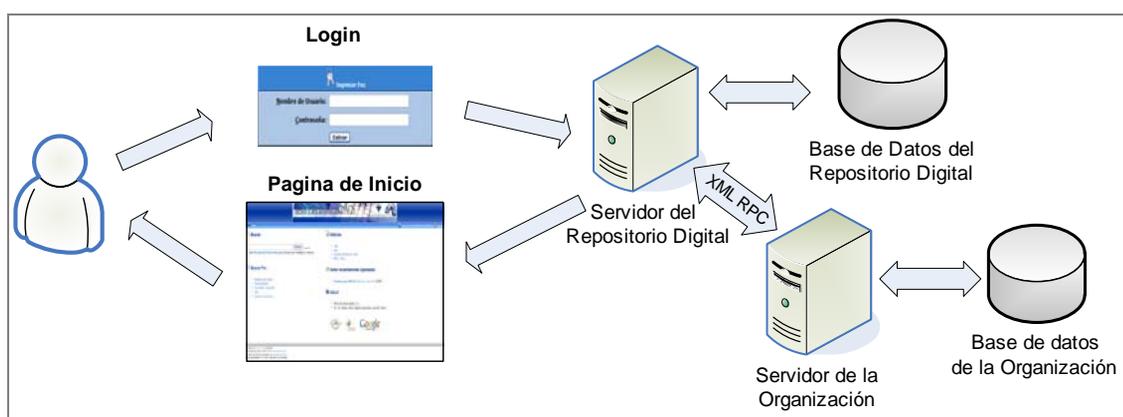


Figura 45: Diagrama de Autenticación de un usuario al Repositorio Digital

Al crear la nueva API para que la autenticación del usuario sea con la base de datos de la organización, solo se deberá editar el Modelo de la arquitectura del sistema, ya que la arquitectura del sistema es Modelo-Vista-Controlador, es transparente el cambio para la vista y el controlador.

En las siguientes figuras se muestran las clases que interactúan al crear un nuevo usuario en el sistema, en la Figura 46 se muestra la interacción del sistema con su propia base de datos, y en la Figura 47 se muestra la clase creada para que interactúe el sistema con la base de datos de la organización.

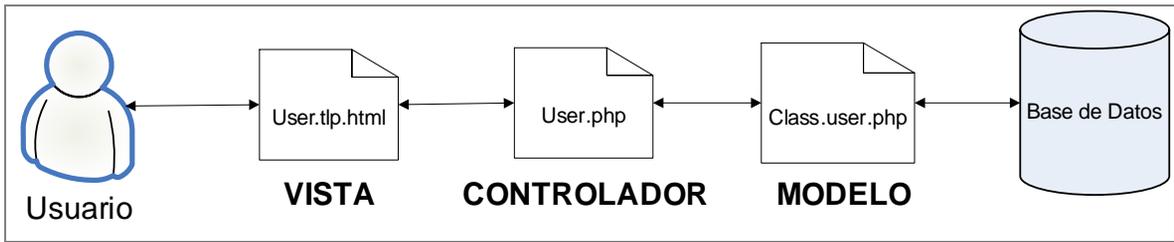


Figura 46: Diagrama de interacción del sistema con su propia base de datos al ingresar un nuevo usuarios al sistema.

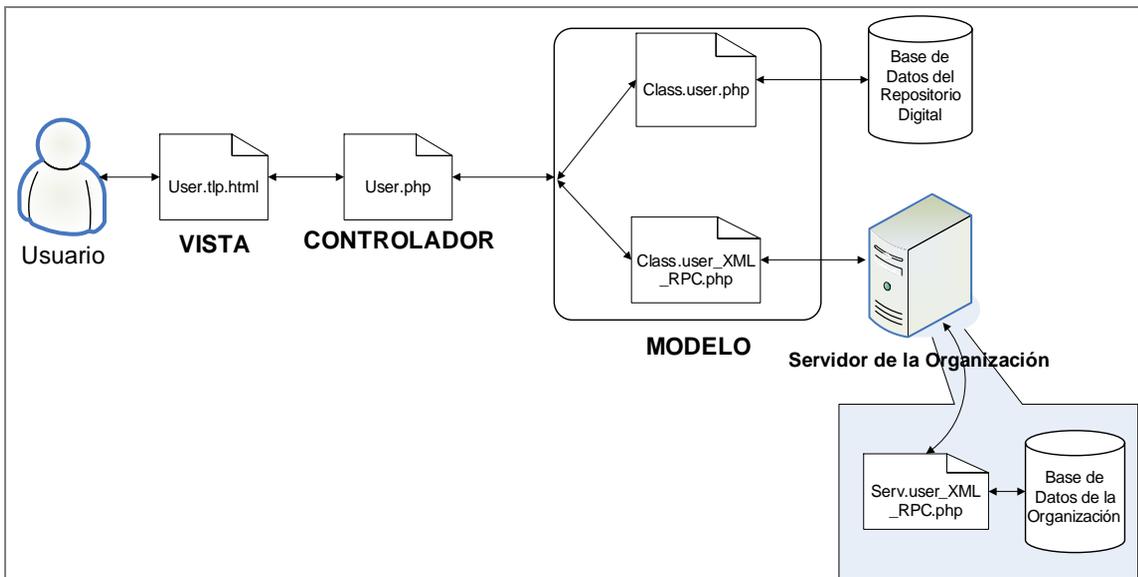


Figura 47: Diagrama de interacción del sistema con la base de datos de la organización al ingresar un nuevo usuario al sistema.

## 6. EVALUACION DEL IMPACTO DE LA IMPLEMENTACION DEL REPOSITORIO DIGITAL EN UNA ORGANIZACIÓN.

En primera instancia la evaluación del repositorio digital se realizaría en la Universidad Austral de Chile, donde se necesitaba que el sistema este funcionando completamente en la plataforma de la Universidad Austral de Chile, para que pueda ser evaluado correctamente su impacto en está. Esto requería de bastante tiempo, por lo tanto, se optó por realizar la evaluación del repositorio digital en la empresa FVT Servicios Telemáticos, donde el sistema gestionará toda la documentación de la empresa.

A continuación se describirá brevemente la configuración de las comunidades, colecciones y roles de seguridad para satisfacer las necesidades de la empresa.

### 6.1 Comunidades Creadas en la Organización

Según las especificaciones de la empresa se crearon las siguientes comunidades y colecciones:

1. Área de Gerencia de FVT Servicios Telemáticos.
2. Área de Finanzas de FVT Servicios Telemáticos.
3. Área de Ingeniería de FVT Servicios Telemáticos.
4. Área de RRHH de FVT Servicios Telemáticos.

Como se muestra en la figura 48.



Figura 48: Comunidades creadas para gestionar los documentos de FVT Ltda...

## **6.2 Colecciones Creadas en la Organización:**

Las colecciones asociadas a la comunidad “Área de Gerencia de FVT Servicios Telemáticos” son:

1. NOA.
2. Informes.
3. Contratos.
4. Documentos de Gerencia.

Las colecciones asociadas a la comunidad “Área de Finanzas de FVT Servicios Telemáticos” son:

1. Presupuestos
2. Facturas.
3. Cotizaciones
4. Balances.
5. Documentos de Finanzas.

Las colecciones asociadas a la comunidad “Área de Ingeniería de FVT Servicios Telemáticos” son:

1. Proyectos.
2. Documentos Externos.
3. Documentos de Procedimientos.
4. Documentos de Informes de Viajes.

Las colecciones asociadas a la comunidad “Área de RRHH de FVT Servicios Telemáticos” son:

1. Vacaciones.
2. Tutoriales.
3. Manuales de Seguridad.

## **6.3 Políticas de Seguridad en la Organización.**

Se crearon grupos de usuarios para cada comunidad, donde cada grupo de usuarios contiene los usuarios autorizados para poder ver, editar, borrar e ingresar archivos a su correspondiente comunidad, en el cual, existen usuarios que pertenecen a mas de un grupo.

## 6.4 Cuestionario de Evaluación

Para evaluar el repositorio digital en FVT Servicios Telemáticos, se desarrollo una serie de preguntas, las cuales, fueron respondidas por la persona encargada de evaluar el sistema. El tiempo de evaluación del sistema fue de 1 mes, en el cual, se ingreso una serie de documentos al sistema en todas las comunidades creadas, para así, evaluar de mejor manera el funcionamiento del repositorio digital.

El cuestionario cuenta con una serie de preguntas donde las respuestas son según el nivel de funcionamiento del sistema. Los niveles de respuesta son del rango del 1 al 5, siendo el nivel 1 el peor nivel de funcionamiento y el 5 el mejor.

1. ¿Es fácil de comprender el modo de uso del sistema?

5

2. ¿Es fácil navegar por los distintos módulos del sistema?

4

3. ¿Cual es el nivel de rapidez de despliegue de la información en el sistema?

3

4. ¿Cuál es el nivel de Búsqueda Simple del sistema?

5

5. ¿Cuál es el nivel de Búsqueda Avanzada del sistema?

4

6. ¿Cual es el nivel de funcionalidad al administrar los documentos por comunidades y colecciones?

5

7. ¿Cuál es el nivel de centralización de la documentación en FVT, que se logró al utilizar el repositorio digital?

3

**Observación:** El nivel alcanzado es principalmente por la descentralización de los documentos que existe actualmente en FVT, ya que los documentos están repartidos en casi todos los computadores de la empresa, además por la carga de trabajo existente en la empresa, en el tiempo de evaluación del sistema.

8. ¿Cuál es el nivel de seguridad de los usuarios con respecto a las distintas colecciones, donde cada grupo de usuario tiene permiso sobre ciertas colecciones (Área de RRHH, Área de Gerencia,..etc.)?

5

9. ¿Cual es el nivel de administración de la documentación de los proyectos de FVT en el sistema?

3

10. ¿Cuál es el nivel de funcionalidad de las Herramientas de Administración (usuarios, grupo de usuarios, noticias, creación de metadatos, índices,...etc.) del sistema?

4

11. ¿Cual es el nivel de Información que entrega el sistema sobre los documentos ingresados (mediante los metadatos)?

5

12. ¿Cuáles son los principales problemas del actual sistema de gestión de los documentos en FVT?

Los principales problemas de la gestión de la documentación en FVT son la desagrupación de los documentos, falta de políticas de seguridad para los usuarios, dispersión, falta de herramientas de gestión de documentos, no existen políticas de acceso a la información.

#### **Otras Observaciones del sistema.**

- Sistema util, facil de usar, con funcionalidades necesarias para la gestion documental de una organización. Sin embargo para su correcta implementación es necesario definir correcta y detalladamente las políticas de seguridad, acceso y definición de los metadatos, para representar un gran porcentaje de los documentos tipo que se manejan tanto en la gestión de proyectos como en la administración de la organización.

## **7. PAUTAS PARA LA IMPLEMENTACION DE UN REPOSITORIO DIGITAL**

Las pautas o recomendaciones para la instalación de un repositorio digital en una organización son las siguientes:

1. Instalar todas las librerías y programas que requiere el repositorio digital.
2. Instalar el repositorio digital Fedora 2.1.1 o una versión superior.
3. Instalar el front-end Fez para el repositorio digital Fedora.
4. Identificar las distintas áreas de la organización que se representarán como comunidades en el repositorio digital.
5. Identificar las distintas colecciones que estarán relacionadas con una o varias comunidades.
6. Definir los distintos tipos de documentación que se ingresarán al repositorio digital.
7. Definir los tipos de documentación que se ingresarán en cada colección.
8. Definir un conjunto de campos para cada tipo de documento que se ingresará en el repositorio digital, donde dicho conjunto de campos, serán las especificaciones más importantes del documento. (Ej: Para un documento de Tesis, unos de los campos son: Título, Auto(es), Patrocinante(s), Facultad, Escuela, etc.)
9. Definir los índices de búsqueda que serán indexados a los campos que se desearán buscar, en búsqueda simple o avanzada (Ej. Autor, Patrocinante, Título, etc.).
10. Definir el estándar de metadatos que se utilizará para ingresar los distintos tipos de documentos, donde el mejor estándar debido a su flexibilidad y generalidad es el MODS.
11. Para la autenticación de los usuarios del sistema se deberá definir si se utilizará la base de datos de la organización (Donde ya se encuentran los usuarios) o la base de datos del sistema (Donde se deberá ingresar todos los usuarios de la organización que interactuarán con el sistema).
12. Definir los usuarios que van a interactuar con el sistema.
13. Definir los grupos de usuarios del sistema.
14. Definir los roles de seguridad de cada comunidad o colección.

## **8. CONCLUSIONES.**

En el proceso de desarrollo de este trabajo de título fue posible identificar y describir tecnologías, tanto para la definición de metadatos, como para la implementación a nivel de software de sistemas capaces de almacenar en forma persistente, y segura la información documental de la más variada índole.

Las tecnologías que se describen en este trabajo son aplicables hoy en día en organizaciones como bibliotecas o empresas de ingeniería. Esto se comprobó mediante entrevistas y realizando pruebas en terreno con muy buenos resultados.

Mediante entrevistas con encargados de la Biblioteca de la Universidad Austral de Chile, se logró implementar una versión de prueba del repositorio digital, para gestionar los documentos de tesis de la universidad. Además se adquirió valiosos conocimientos sobre estándares de metadatos para la publicación de documentos de tesis, y sobre la utilización de comunidades y colecciones para la gestión de documentos en una organización.

En el proceso de desarrollo de la implementación del repositorio digital, se crearon pautas detalladas, donde se explica paso a paso la instalación, y configuración de los distintos softwares y librerías necesarias para la correcta implementación del repositorio digital.

En el marco de cooperación con el proyecto Fez/Fedora, en este trabajo de título se logró crear una API para la interconexión entre el sistema (Fez/Fedora) y la base de datos de la empresa, donde se implementará el repositorio digital, así no tendrán que ser migrados todos los usuarios de la empresa a la base de datos del sistema para la autenticación. Esto permite una mayor flexibilidad del sistema, asegurando la privacidad de la información de los usuarios, y la centralización de la información en la institución u organización.

Con la implementación de los repositorio digitales que se realizaron en este trabajo de título (para gestionar los documentos de tesis y los de FVT Servicios Telemáticos), fue posible adquirir conocimientos suficientes para desarrollar pautas que permitirán la implementación de un repositorio digital en una institución o organización, indicando todos los pasos que se deben tener en cuenta para realizar una correcta implementación.

En el transcurso de este trabajo de título se logró implementar el repositorio digital en dependencias de FVT Ltda., en donde ingenieros de dicha empresa utilizaron el sistema para gestionar su documentación interna, luego se realizaron entrevistas para poder rescatar su experiencia al utilizar el sistema y así poder evaluar el funcionamiento del repositorio digital.

## 9. REFERENCIAS BIBLIOGRAFICAS.

- [BAR02] José Manuel Barrueco (2002).OAI-PMH: Protocolo para la transmisión de contenidos en Internet. Biblioteca de Ciencias Sociales de la Universidad de Valencia. Consultado el 20 de Mayo de 2008
- [BENnd] Henry Benítez & Stella Robayo. Protocolo Z39.50 Una Herramienta Importante en la Recuperación de la Información. Universidad de la Salle. Consultado el 2 de Marzo de 2008
- [BTR01] Biblio Tech Review (2001). Information Technology for Libraries. Disponible en [http://www.biblio-tech.com/html/z39\\_50.html](http://www.biblio-tech.com/html/z39_50.html). Consultado el 25 de Junio de 2008
- [CDS] CDS Invenio Overview. Disponible en <http://cdsware.cern.ch>. Consultado el 10 de Noviembre de 2007.
- [Cli03] Clifford A. Lynch, (2003). Institutional Repositories: Essential Infrastructure for Scholarship in the Digital Age.
- [DCM08] Dublin Core Metadata Initiative (2008). Disponible en <http://dublincore.org/> Consultado el 20 de Marzo de 2008
- [DSP] Sitio Oficial del Repositorio digital Dspace. Disponible en <http://www.dspace.org>. Consultado el 10 de Noviembre de 2007.
- [EPR] Sitio Oficial del Repositorio Digital E-Print. Disponible en <http://eprints.rclis.org>. Consultado el 10 de Noviembre de 2007.
- [ECH04] Garikoits Echevarria Contreras (2004) Presentación OAI-PMH, proyecto OAC. Disponible en [www.deli.deusto.es/Resources/Documents/OAI-PMH\\_presentacion.ppt](http://www.deli.deusto.es/Resources/Documents/OAI-PMH_presentacion.ppt). Consultado el 20 de Marzo de 2008
- [FDT05a] The Fedora Development Team (2005).Tutorial: Introduction to Fedora. Disponible en [www.Fedora.info/download/2.0/userdocs/tutorials/tutorial1.pdf](http://www.Fedora.info/download/2.0/userdocs/tutorials/tutorial1.pdf) Consultado el 10 de Noviembre de 2007.
- [FDT05b] The Fedora Development Team (2005). Fedora Open Source Repository Software: White Paper. Disponible en [www.Fedora.info/pdfs/WhitePaper.10.28.05.pdf](http://www.Fedora.info/pdfs/WhitePaper.10.28.05.pdf). Consultado el 10 de Noviembre de 2007.
- [FED] Sitio Oficial del Repositorio Digital Fedora. Disponible en <http://www.Fedora.info>. Consultado el 10 de Noviembre de 2007
- [FEZ] Wiki del proyecto Fez.

- Disponible en [http://dev-repo.library.uq.edu.au/wiki/index.php/Main\\_Page](http://dev-repo.library.uq.edu.au/wiki/index.php/Main_Page)  
Consultado el 5 de Diciembre de 2007.
- [FUE07] Gema Bueno de la Fuente (2007). Introducción a los Metadatos: Aplicación de los Metadatos en el ámbito Bibliotecario.
- [INE03] Instituto Nacional De Estadística, Geografía e Informática (2003). Presentación sobre la descripción de metadatos. Disponible en <http://antares.inegi.gob.mx/metadatos/metadat1.ppt>. Consultado el 20 de Enero de 2007.
- [ITO] Sitio Oficial del repositorio I-Tor. Disponible en [www.i-tor.org](http://www.i-tor.org). Consultado el 10 de Noviembre de 2007.
- [KOR06] Christiaan Kortekaas (2006). Don't Keep it Under Your Hat. The University of Queensland, Australia. Conference Paper for The Fedora Users Conference 2006, Virginia USA.
- [LOC99a] The Library Of Congress- Network Development and MARC standards Office. Disponible en <http://www.loc.gov/marc/>. Consultado el 20 de Enero de 2007.
- [LOC b] The Library Of Congress-Standards Metadata Object Description Schema. Disponible en <http://www.loc.gov/standards/mods/>. Consultado el 18 de Febrero de 2007.
- [LOP07] Alicia López Medina (2007). Guía para la puesta en marcha de un Repositorio institucional, 29 de marzo del 2007.
- [NWI] Norma Web Info Z29.50. Disponible en <http://www.bcl.jcyl.es/zeta/>  
Consultado el 5 de Diciembre de 2007.
- [OAF] Open Archives Forum. Pagina que contiene tutoriales, workshops proyectos. Disponible en <http://www.oaforum.org/tutorial/english/page3.htm>  
Consultado el 15 de Marzo de 2008.
- [OAI] Open Archives Initiative. Disponible en <http://www.openarchives.org/>.  
Consultado el 15 de Marzo de 2008
- [PAY05] Sandy Payette (2005), Co-Director Fedora Project, Cornell University. Fedora: New Features, New Collaborations, Bright Future.  
Disponible en [www.deli.deusto.es/Resources/Documents/OAI-PMH\\_presentacion.ppt](http://www.deli.deusto.es/Resources/Documents/OAI-PMH_presentacion.ppt). Consultado el 15 de Marzo de 2008

- [SMR] Pagina oficial de Smarty. Disponible en <http://www.smarty.net/> Consultado el 15 de Abril de 2008
- [UNE05] Biblioteca de la UNED (Universidad Nacional de Educacion a Distancia). Proyecto “e-Espacio UNED” Repositorio Institucional de la UNED. Disponible en:  
<http://e-spacio.uned.es/Fez/eserv.php?pid=bibliuned:346&dsID=Memoriae-spaciodefinitiva.doc>. Consultado el 18 de Mayo de 2008
- [W3C04a] World Wide Web Consortium. This page summarizes the work of RDF Core Working Group, which was completed in 2004. Disponible en <http://www.w3.org/RDF/>. Consultado el 20 de Mayo de 2008
- [W3C04b] World Wide Web Consortium. XML Schema. Disponible en <http://www.w3.org/TR/xmlschema-0/>. Consultado el 20 de Mayo de 2008