



Proyecto de Grado:

Presentado ante la ilustre UNIVERSIDAD DE LOS ANDES como requisito parcial para  
obtener el Título de INGENIERO DE SISTEMAS

**IMPLEMENTACIÓN DE UN CONVERTOR DE MECANISMOS DE  
CONSULTAS Y ACCESOS ENTRE SISTEMAS DE REPOSITARIOS  
INSTITUCIONALES.  
CASO ALEJANDRÍA-DSPACE.**

Por

Br. Tupac Bastidas

Tutora: Hilda Contreras

Asesor industrial: Rodrigo Torréns

Junio 2008

©2008 Universidad de Los Andes Mérida, Venezuela

## **Implementación de un conversor de mecanismos de consultas y accesos entre sistemas de repositorios institucionales. Caso Alejandría-DSpace.**

Br. Tupac Amaru Bastidas Celis

### **Resumen del proyecto:**

El repositorio institucional SABER-ULA, que desde el año 2000 preserva y difunde la producción intelectual de la Universidad de Los Andes a través de Internet, está ubicado entre los primeros cuarenta repositorios institucionales a nivel mundial; referenciado desde diferentes índices, buscadores, bibliotecas y páginas Web. Actualmente, motivado por el acceso abierto al conocimiento y la promoción del uso del software libre, este repositorio está siendo migrado desde la plataforma privativa Alejandría al sistema de repositorio institucional de licencia libre DSpace. Esta tesis considera los diferentes factores que permitirán mantener, de forma automática y transparente para los usuarios, los enlaces de accesos y consultas, desde cualquier dominio, hacia los datos del repositorio ubicados en la nueva plataforma. Por tanto, se plantea la necesidad de un “conversor” que permita traducir o buscar los accesos y consultas equivalentes de uno a otro sistema y además adecuarlos al sistema destino. Esto se pretende lograr implementando un componente de software configurable que convierta las solicitudes de información hacia el servicio de SABER-ULA en una respuesta automática de la plataforma DSpace, a través de la conveniente traducción e interpretación de la identificación de los registros de información y de los diferentes esquemas de consultas de ambos sistemas.

### **Palabras claves:**

Acceso Abierto al Conocimiento, Repositorios Institucionales, Metadatos, Sistemas de Identificación de Documento, DSpace.

*A mi madre, por su amor incondicional  
y eterno apoyo en cada momento de mi vida.*

*Al amor de mi vida; Astrid, por llenar mi vida  
de alegría, ternura, felicidad y amor. ¡Gracias!*

## Agradecimientos

A mi familia: especialmente mi padre, mi hermana, mi abuela, mis tíos. Gracias por formar parte de este proyecto de vida y por su desinteresada colaboración desde el inicio hasta la finalización del mismo.

A la profesora Hilda Contreras, por su apoyo, motivación y tutoría durante los meses de desarrollo de este proyecto. Gracias por asumir este reto y permitirme ser su primer tesista.

Al CTI-CPTM, particularmente a mis compañeros en SABER-ULA: Mi jefe Rodrigo, Maye, Susana, Carol, Henry, Ruth, Lady, Zulay, Pili, Eddy y Juan. Les estaré eternamente agradecido por abrirme las puertas de su unidad, prestarme su apoyo y regalarme una grata experiencia laboral.

A la profesora Flor y al profesor Rafael, de no ser por ustedes no estaría hoy imprimiendo estas palabras. Gracias por el grano de arena que colocaron para que esto fuera posible.

A mis amigos, con quienes compartí momentos agradables durante el desarrollo de éste proyecto. Fueron sin excepción motivo para extenderles unas palabras de agradecimiento por su apoyo y amistad.

# Índice

<b>Índice</b> .....	<b>5</b>
<b>Índice de figuras</b> .....	<b>8</b>
<b>Capítulo 1    Introducción</b> .....	<b>11</b>
<b>1.1    Antecedentes</b> .....	<b>12</b>
<b>1.2    Planteamiento del Problema</b> .....	<b>13</b>
<b>1.3    Objetivos</b> .....	<b>16</b>
1.3.1    Objetivo General .....	16
1.3.2    Objetivos Específicos .....	16
<b>1.4    Metodología</b> .....	<b>16</b>
1.4.1    Primera Fase: Investigación.....	16
1.4.2    Segunda Fase: Análisis, diseño e implementación .....	17
1.4.3    Tercera Fase: Instalación y pruebas.....	18
1.4.4    Cuarta Fase: Documentación final de Tesis .....	18
<b>Capítulo 2    Marco Teórico</b> .....	<b>19</b>
<b>2.1    Iniciativas de Acceso Abierto al Conocimiento</b> .....	<b>19</b>
2.1.1    Iniciativa de Acceso Abierto de Budapest.....	20
2.1.2    Declaración de Berlín sobre Acceso Abierto al Conocimiento .....	21
<b>2.2    Tecnologías de Acceso Abierto</b> .....	<b>22</b>
2.2.1    Estándares de Interfaz y Contenido .....	22
World Wide Web .....	22
Métodos de envío de peticiones .....	23
Metadatos .....	24
2.2.2    Estándares o Mecanismos de Identificación de Documentos .....	25
Identificadores de Objetos Digitales .....	26
2.2.3    Herramientas de Publicación Electrónica.....	27
¿Que es un Repositorio Institucional (RI)? .....	27
El Repositorio Institucional (RI) SABER-ULA.....	28
Plataformas para la Publicación Electrónica .....	29
<b>2.3    Herramientas de redireccionamiento de URL</b> .....	<b>31</b>
Apache Alias .....	31
Apache mod_rewrite .....	32
Etiqueta Meta Refresh.....	33
JavaScript window.location .....	34

PHP: header().....	35
<b>Capítulo 3   Análisis.....</b>	<b>37</b>
Morfología del URL.....	37
<b>3.1    Estudio del sistema de consultas.....</b>	<b>38</b>
3.1.1    Sistema de consultas de Alejandría .....	38
3.1.2    Sistema de consultas de DSpace.....	41
<b>3.2    Estudio del sistema de direccionamiento de documentos.....</b>	<b>43</b>
3.2.1    Sistema de direccionamiento de documentos de Alejandría.....	43
3.2.2    Sistema de direccionamiento de documentos de Dspace.....	46
<b>3.3    Comparación entre los sistemas de repositorios institucionales.....</b>	<b>48</b>
3.3.1    Comparación entre los sistemas de consultas .....	48
3.3.2    Comparación entre los sistemas de direccionamiento de documentos .....	50
<b>Capítulo 4   Diseño.....</b>	<b>52</b>
<b>4.1    Clasificación de las solicitudes Web.....</b>	<b>52</b>
4.1.1    Método GET.....	52
4.1.2    Método POST.....	54
<b>4.2    Diseño del conversor.....</b>	<b>55</b>
4.2.1    Diseño de las aplicaciones de redireccionamiento de solicitudes Web .....	57
Redireccionar consultas .....	58
Redireccionar accesos .....	59
Redireccionar documentos .....	60
4.2.2    Diseño la de configuración y generación de las aplicaciones de redireccionamiento del sistema..	61
Instalar conversor.....	63
Configurar conversor .....	64
Manejar la base de datos asociativa .....	65
Generar aplicaciones y reglas de redireccionamiento .....	69
<b>Capítulo 5   Implementación.....</b>	<b>71</b>
<b>5.1    Implementación del sistema conversor.....</b>	<b>71</b>
Aplicaciones de redireccionamiento .....	71
Redireccionamiento .....	72
Base de Datos.....	74
Archivo de configuración de casos de redireccionamiento .....	74
Interfaz gráfica de usuario.....	74
Manual de administración del sistema .....	75
<b>5.2    Aplicación del conversor al caso de estudio.....</b>	<b>76</b>
<b>5.3    Refinamiento por versiones y pruebas.....</b>	<b>78</b>

5.3.1	Versiones del sistema conversor.....	79
	Versiones 1 a 5.....	79
	Versiones 6 a 9.....	79
	Versiones 10 a 12.....	80
5.3.2	Pruebas .....	80
	Pruebas de instalación.....	80
	Pruebas de redireccionamiento .....	80
	Pruebas de verificación de enlaces rotos.....	83
	Pruebas de rendimiento.....	85
	<b><i>Conclusiones</i></b> .....	<b>89</b>
	<b><i>Anexos</i></b> .....	<b>92</b>
	<b><i>Bibliografía y Referencias</i></b> .....	<b>112</b>

# Índice de figuras

FIGURA 1. CONSULTAS DEL RI SABER-ULA. TOMADO DE <a href="http://www.saber.ula.ve/estadisticas">HTTP://WWW.SABER.ULA.VE/ESTADISTICAS</a> .....	11
FIGURA 2. ESQUEMA DE FUNCIONAMIENTO DEL SERVICIO PURL. TOMADO DE <a href="http://purl.oclc.org/docs/new_purl_summary.html">HTTP://PURL.OCLC.ORG/DOCS/NEW_PURL_SUMMARY.HTML</a> .....	13
FIGURA 3. PORCENTAJE DE LOS INDICADORES DEL RANKING MUNDIAL DE UNIVERSIDADES EN LA RED. TOMADO DE <a href="http://www.webometrics.info">HTTP://WWW.WEBOMETRICS.INFO</a> .....	14
FIGURA 4. UBICACIÓN DE SABER-ULA EN EL TOP 200 REPOSITORIOS DEL RANKING MUNDIAL DE UNIVERSIDADES EN LA RED. TOMADO DE <a href="http://www.webometrics.info/top200_rep.asp">HTTP://WWW.WEBOMETRICS.INFO/TOP200_REP.ASP</a> .....	14
FIGURA 5. MODELO DEL PROBLEMA DE ACCESO AUTOMÁTICO DE LOS REQUERIMIENTOS AL SERVIDOR SABER-ULA A TRAVÉS DEL “CONVERSOR” LUEGO DE LA MIGRACIÓN.....	15
FIGURA 6. LISTADO DE LOS ELEMENTOS DEL DUBLÍN CORE. ....	25
FIGURA 7. TABLA DE LAS VARIABLES DE BÚSQUEDA MÁS USADAS POR EL SISTEMA ALEJANDRÍA. ....	39
FIGURA 8. CAJA DE BÚSQUEDA DEL SERVIDOR SABER-ULA. TOMADO DE <a href="http://www.saber.ula.ve/unidades/investigadores.html">HTTP://WWW.SABER.ULA.VE/UNIDADES/INVESTIGADORES.HTML</a> .....	40
FIGURA 9. PÁGINA RESULTANTE DE UNA CONSULTA A SABER-ULA. TOMADO DE <a href="http://www.saber.ula.ve/cgi-win/be_alex.exe?AUTOR=ACOSTA+KANQUIS,+NEBIS">HTTP://WWW.SABER.ULA.VE/CGI-WIN/BE_ALEX.EXE?AUTOR=ACOSTA+KANQUIS,+NEBIS</a> .....	40
FIGURA 10. CAJA DE BÚSQUEDA DE DSPACE. TOMADO DE <a href="http://dspace.mit.edu/advanced-search">HTTP://DSPACE.MIT.EDU/ADVANCED-SEARCH</a> .....	41
FIGURA 11. TABLA DE LAS VARIABLES DE BÚSQUEDA DEL SISTEMA DSPACE. ....	41
FIGURA 12. PÁGINA RESULTANTE DE UNA CONSULTA A DSPACE. TOMADO DE <a href="http://dspace.mit.edu/simple-search?query=((author:john))">HTTP://DSPACE.MIT.EDU/SIMPLE-SEARCH?QUERY=((AUTHOR:JOHN))</a> .....	43
FIGURA 13. TABLA DE LOS IDENTIFICADORES DE REGISTRO DEL LIBRO “SISTEMAS DE CONEXIÓN A TIERRA EN 60 HZ” EN EL SERVIDOR SABER-ULA. ....	44
FIGURA 14. TABLA DE LAS VARIABLES DE ACCESO DIRECTO A DOCUMENTOS DE ALEJANDRÍA .....	45
FIGURA 15. PÁGINA RESULTANTE DE LA SOLICITUD DE UN REGISTRO A SABER-ULA. TOMADO DE <a href="http://www.saber.ula.ve/cgi-win/be_alex.exe?ACCESO=T016300001447/0">HTTP://WWW.SABER.ULA.VE/CGI-WIN/BE_ALEX.EXE?ACCESO=T016300001447/0</a> .....	46
FIGURA 16. PÁGINA RESULTANTE DE LA SOLICITUD DE UN REGISTRO A DSPACE. TOMADO DE <a href="http://dspace.mit.edu/handle/1721.1/30452">HTTP://DSPACE.MIT.EDU/HANDLE/1721.1/30452</a> .....	47
FIGURA 17. TABLA COMPARATIVA DE LOS PARES (VARIABLE-VALOR) DE CONSULTA ENTRE LOS SISTEMAS DE REPOSITORIOS INSTITUCIONALES. ....	48
FIGURA 18. TABLA ASOCIATIVA ENTRE LAS VARIABLES DE CONSULTA DE ALEJANDRÍA Y DSPACE. ....	49
FIGURA 19. TABLA COMPARATIVA DE LAS CARACTERÍSTICAS PRINCIPALES DE LOS SISTEMAS DE CONSULTA DE ALEJANDRÍA Y DSPACE.....	49
FIGURA 20. TABLA COMPARATIVA DE LA COMPOSICIÓN DE LOS IDENTIFICADORES DE REGISTROS DE INFORMACIÓN EN LOS SISTEMAS ALEJANDRÍA Y DSPACE.....	50
FIGURA 21. TABLA COMPARATIVA DE LAS CARACTERÍSTICAS PRINCIPALES DE LOS SISTEMAS DE DIRECCIONAMIENTO DE DOCUMENTOS DE ALEJANDRÍA Y DSPACE. ....	51
FIGURA 22. CLASIFICACIÓN DE LAS SOLICITUDES WEB.....	52
FIGURA 23. DIAGRAMA FUNCIONAL DEL CONVERSOR.....	55
FIGURA 24. DIAGRAMA DE COMPONENTES DEL SISTEMA CONVERSOR. ....	56



FIGURA 25. ESTRUCTURA DE UN SISTEMA DE REPOSITORIO INSTITUCIONAL. ....	56
FIGURA 26. ESTRUCTURA DEL CONVERSOR.....	57
FIGURA 27. TABLA DE ACTORES Y CASOS DE USO DEL DISEÑO DE LAS APLICACIONES DE REDIRECCIONAMIENTO... ..	58
FIGURA 28. CASO DE USO “REDIRECCIONAR CONSULTA”.....	58
FIGURA 29. DIAGRAMA DE ACTIVIDADES DEL CASO DE USO “REDIRECCIONAR CONSULTA”.....	59
FIGURA 30. CASO DE USO “REDIRECCIONAR ACCESO”.....	59
FIGURA 31. DIAGRAMA DE ACTIVIDADES DEL CASO DE USO “REDIRECCIONAR ACCESO”.....	60
FIGURA 32. CASO DE USO “REDIRECCIONAR DOCUMENTO”.....	61
FIGURA 33. DIAGRAMA DE ACTIVIDADES DEL CASO DE USO “REDIRECCIONAR DOCUMENTO”.....	61
FIGURA 34. DISEÑO GENERAL DE LA INTERFAZ GRÁFICA DE USUARIO.....	62
FIGURA 35. TABLA DE ACTORES Y CASOS DE USO DEL DISEÑO DE LA CONFIGURACIÓN Y GENERACIÓN DE LAS APLICACIONES DE REDIRECCIONAMIENTO.....	63
FIGURA 36. CASO DE USO “INSTALAR CONVERSOR”.....	63
FIGURA 37. DIAGRAMA DE ACTIVIDADES DEL CASO DE USO “INSTALAR CONVERSOR”.....	64
FIGURA 38. CASO DE USO “CONFIGURAR CONVERSOR”.....	64
FIGURA 39. DIAGRAMA DE ACTIVIDADES DEL CASO DE USO “CONFIGURAR CONVERSOR”.....	65
FIGURA 40. CARACTERÍSTICAS ASOCIATIVAS PARA EL SISTEMA DE RI ORIGEN (ALEJANDRÍA).....	65
FIGURA 41. ESQUEMA DE LA ASOCIACIÓN DE UN ACCESO (ALEJANDRÍA) CON UN HANDLE (DSpace).....	66
FIGURA 42. TABLA ASOCIATIVA DE LAS CARACTERÍSTICAS ENTRE LOS RRII.....	66
FIGURA 43. DISEÑO CONCEPTUAL DE LA BASE DE DATOS ASOCIATIVA.....	67
FIGURA 44. DISEÑO IMPLEMENTABLE DE LA BASE DE DATOS.....	67
FIGURA 45. CASO DE USO “MANEJAR BD ASOCIATIVA”.....	68
FIGURA 46. ESQUEMA ENTRADA-SALIDA DEL MÓDULO DE MANEJO DE LA BD ASOCIATIVA.....	68
FIGURA 47. DIAGRAMA DE ACTIVIDADES DEL CASO DE USO “MANEJAR BD ASOCIATIVA”.....	69
FIGURA 48. CASO DE USO “GENERAR APLICACIONES Y REGLAS DE REDIRECCIONAMIENTO”.....	69
FIGURA 49. ESQUEMA ENTRADA-SALIDA DEL MÓDULO DE GENERACIÓN DE APLICACIONES Y REGLAS DE REDIRECCIONAMIENTO.....	70
FIGURA 50. DIAGRAMA DE ACTIVIDADES DEL CASO DE USO “GENERAR APLICACIONES Y REGLAS DE REDIRECCIONAMIENTO”.....	70
FIGURA 51. TABLA COMPARATIVA DE LAS CARACTERÍSTICAS FUNCIONALES DE LAS HERRAMIENTAS DE DIRECCIONAMIENTO (ALIAS, MOD_REWRITE, “REFRESH”, HEADER()) Y JAVASCRIPT).....	73
FIGURA 52. INTERFAZ GRÁFICA DEL CONVERSOR.....	75
FIGURA 53. ESQUEMA DE LA APLICACIÓN DEL CONVERSOR AL CASO DE ESTUDIO.....	77
FIGURA 54. TABLA DE LOS CASOS IDENTIFICADOS PARA EL CASO DE ESTUDIO ALEJANDRÍA-DSpace.....	77
FIGURA 55. CONFIGURACIÓN DE LAS TABLAS ASOCIATIVAS PARA EL CASO DE ESTUDIO ALEJANDRÍA-DSpace.....	78
FIGURA 56. PÁGINA DE PRUEBAS DE REDIRECCIONAMIENTO PARA EL CASO DE ESTUDIO ALEJANDRÍA-DSpace.....	81
FIGURA 57. TABLA DE RESULTADOS DE LAS PRUEBAS DE CAJA NEGRA AL CONVERSOR.....	82
FIGURA 58. RESULTADO DE UNA PRUEBA DE CAJA NEGRA A UNA CAJA DE BÚSQUEDA.....	82
FIGURA 59. RESULTADO DE UNA PRUEBA DE CAJA BLANCA APLICADA AL CONVERSOR.....	83
FIGURA 60. RESULTADO DE LA VERIFICACIÓN DE ENLACES ROTOS AL SERVIDOR SABER-ULA CON EL CONVERSOR DESINSTALADO.....	84

FIGURA 61. RESULTADO DE LA VERIFICACIÓN DE ENLACES ROTOS AL SERVIDOR SABER-ULA CON EL CONVERSOR INSTALADO. ....	84
FIGURA 62. RESULTADO DEL BENCHMARKING AL SERVIDOR APACHE DETENIDO .....	85
FIGURA 63. RESULTADO DEL <i>BENCHMARKING</i> AL SERVIDOR APACHE INICIADO Y MANEJADOR DE BBDD DETENIDO. .....	86
FIGURA 64. RESULTADO DEL <i>BENCHMARKING</i> AL SERVIDOR APACHE INICIADO Y MANEJADOR DE BBDD FUNCIONANDO. ....	86
FIGURA 65. RESULTADO DEL <i>BENCHMARKING</i> AL SERVIDOR APACHE INICIADO, CONVERSOR INSTALADO Y SERVIDOR TOMCAT DETENIDO. ....	87
FIGURA 66. RESULTADO DEL <i>BENCHMARKING</i> AL SERVIDOR APACHE INICIADO, CONVERSOR INSTALADO Y SERVIDOR TOMCAT INICIADO. ....	87

## Capítulo 1 Introducción

La Universidad de Los Andes (ULA), en consonancia con su papel protagónico en la construcción de un ambiente propicio para la innovación tecnológica, desde el año 2000 se propuso preservar y difundir la producción y el patrimonio intelectual universitario. Esto se logró gracias a la construcción de un Repositorio Institucional (RI) conocido como SABER-ULA, el cual trabaja bajo un sistema privativo llamado Alejandría, adaptado para soportar repositorios institucionales (Dávila et al, 2006).

SABER-ULA está actualmente situado como el RI número 23 en el mundo según el *World ranking on the Web* (Webometrics, 2008), lo que lo hace acreedor de una presencia importante en Internet entre las herramientas que promueven el acceso abierto al conocimiento. Hoy en día, existen diferentes entes que hacen referencia al repositorio: índices, buscadores, bibliotecas, páginas institucionales, etc. Hasta febrero del 2008 el repositorio superó 28 millones de visitas, y la tendencia es creciente, esto indica la importancia para el mundo académico de la difusión de la producción intelectual de la ULA a través del repositorio. La figura 1, muestra el comportamiento de las consultas hacia el servidor desde su creación hasta la fecha actual.

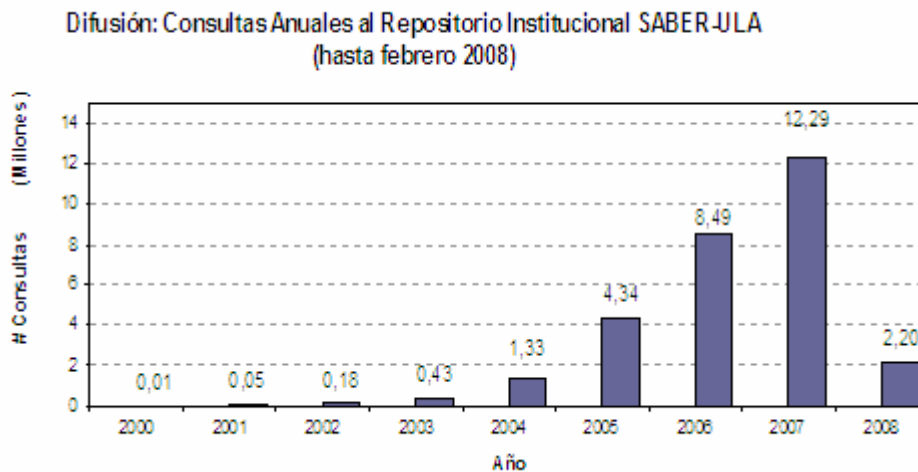


Figura 1. Consultas del RI SABER-ULA. Tomado de <http://www.saber.ula.ve/estadisticas>

Las estadísticas de acceso y consulta al servidor han demostrado un promedio actual de descargas de documentos de 9 Gbyte por día. El origen de estas consultas es variado: alrededor de 10% provienen de la propia Universidad (ULA), el 30% provienen de otros lugares de

Venezuela; un 20% provienen de países Iberoamericanos y Norteamérica. El resto (40%) provienen de otros continentes (especialmente Europa) y países alrededor del mundo, y de robots de búsqueda<sup>1</sup>.

Por otra parte, el gobierno nacional decretó en el 2004 que todos los órganos y entes de la Administración Pública Nacional deben emplear prioritariamente plataformas de software libre (Gaceta Oficial de la República Bolivariana de Venezuela, 2004). En vista de esto, la ULA ha tratado de cumplir con este decreto y, para transformar los espacios universitarios al mundo del software libre, crea un plan de migración masivo de toda su plataforma, contemplando entre sus metas la migración del repositorio SABER-ULA hacia una plataforma libre (Consejo de Computación Académica, 2006). Para esta migración se ha seleccionado la plataforma DSpace<sup>2</sup>, de licencia libre BSD y creada por los laboratorios HP y el MIT, con las tecnologías y estándares propios del acceso abierto y del software libre, como la más adecuada para la migración del actual repositorio.

## 1.1 Antecedentes

Aun cuando han existido muchos procesos de migración de bases de datos entre distintas plataformas, se encontraron pocas referencias acerca de la migración de un repositorio institucional, en parte, debido a lo reciente de esta iniciativa y las tecnologías que la soportan. Aun así, no se encontraron referencias de que se haya contemplado el problema de la pérdida de los enlaces externos al repositorio, y en consecuencia la pérdida de su visibilidad Web. Sin embargo, de la migración de algunos sitios Web se conoce el uso de las herramientas de redireccionamiento (explicadas en la sección 2.3) aplicadas solamente a la página principal del sitio Web o a un directorio de páginas estáticas dentro de un servidor.

Por otra parte, en el año 1995, el departamento de investigación de la OCLC<sup>3</sup> (Online Computer Library Center) desarrolló un servicio de redireccionamiento de URL<sup>4</sup> denominado PURL<sup>5</sup> (Persistent Uniform Resource Locator). Un PURL es un URL que en vez de apuntar directamente a un recurso, apunta a un servicio de resolución intermediario. Este servicio asocia

---

<sup>1</sup> Se puede observar una representación gráfica del origen de las consultas al RI de la ULA en el siguiente URL: <http://clustrmaps.com/counter/maps.php?url=http://www.saber.ula.ve>

<sup>2</sup> Dspace: <http://www.dspace.org/>

<sup>3</sup> OCLC: <http://www.oclc.org/>

<sup>4</sup> *Uniform Resource Locator*; <http://www.w3.org/Addressing/rfc1738.txt>

<sup>5</sup> PURL: <http://purl.org/>

el PURL con el URL actual del recurso y lo devuelve al cliente, el cual completa la transacción normalmente.

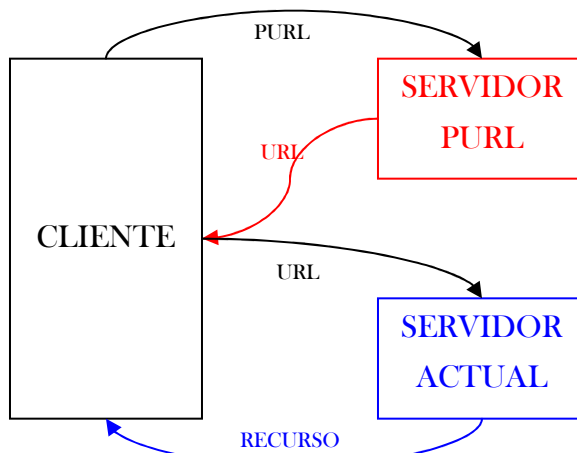


Figura 2. Esquema de funcionamiento del servicio PURL. Tomado de [http://purl.oclc.org/docs/new\\_purl\\_summary.html](http://purl.oclc.org/docs/new_purl_summary.html)

Como se ilustra en la figura 2, el servicio es capaz de redireccionar total o parcialmente un PURL estático, asociarlo con la dirección real del recurso por medio de una consulta a una base de datos encargada de enlazar ambos localizadores, y (en caso de que exista la asociación) devolver el URL real. Adicionalmente, este servicio de localización persistente posee la capacidad de manejar un tipo de identificador de recursos denominado “Handle”, si éste se encuentra asociado a un URL determinado.

## 1.2 Planteamiento del Problema

La Universidad de Los Andes, firmante de la declaración de Berlín<sup>6</sup> en el año 2006 y que además promueve el acceso abierto al conocimiento a través del RI SABER-ULA desde el año 2000, pretende migrar los datos contenidos en este RI hacia una nueva plataforma de software libre. Actualmente se estima que luego del proceso de migración del repositorio hacia la plataforma DSpace, la presencia de la ULA en Internet, así como la difusión del conocimiento de manera abierta, puedan verse muy afectados. Según Webometrics (2008), los indicadores Web reflejan mejor la actividad global de las instituciones ya que existe un número considerable de actividades ejercidas por profesores e investigadores visibles a través de la Web. El primer indicador Web, es el Factor de Impacto Web, que combina el número de enlaces

<sup>6</sup> Ver glosario (anexo A2).

externos entrantes (visibilidad) con el número de páginas Web de un dominio (tamaño). Además, se consideran también el número de documentos, medido por el número de archivos ricos<sup>7</sup> que contiene un dominio Web, y el número de publicaciones comprendidas en la base de datos del Google Académico<sup>8</sup>.

Justamente, la visibilidad, entendida como el número total de enlaces externos únicos recibidos por un sitio, es considerado por este ranking como uno de los indicadores más importantes, representando el 50% tal como lo muestra la figura 3. El número de enlaces entrantes externos recibidos por un dominio se considera como una medida que representa la visibilidad y el impacto del material publicado. Aunque puede existir una gran diversidad de motivaciones para que existan esos enlaces, parece que una fracción significativa de ellos funciona de forma similar a las citas bibliográficas.

RANKING WEBOMETRICS		
VISIBILIDAD (enlaces entrantes externos)	TAMAÑO (páginas Web)	20%
50%	ARCHIVOS RICOS	15%
	ACADÉMICO	15%

Figura 3. Porcentaje de los indicadores del Ranking Mundial de Universidades en la Red. Tomado de <http://www.webometrics.info>

El sitio Web de SABER-ULA esta ubicado dentro de la categoría de Repositorios Académicos con los datos que muestra la figura 4. Cabe destacar que este sitio presenta un aproximado de 10.300 referencias, de las cuales un 50% son externas al dominio, es decir corresponden a la visibilidad. Esto hace necesario considerar en el proceso de migración, el problema de cómo conservar las referencias hacia el sitio y básicamente el nivel de visibilidad alcanzado para no afectar a la institución.

RANKING MUNDIAL	REPOSITORIO	PAÍS	TAMAÑO	POSICIÓN		
				VISIBILIDAD	FICHEROS RICOS	SCHOLAR
23	UNIVERSIDAD DE LOS ANDES REPOSITORIO INSTITUCIONAL		45	61	84	17

Figura 4. Ubicación de SABER-ULA en el TOP 200 Repositorios del Ranking Mundial de Universidades en la Red. Tomado de [http://www.webometrics.info/top200\\_rep.asp](http://www.webometrics.info/top200_rep.asp)

<sup>7</sup> Los archivos ricos, son aquellos archivos relevantes en las actividades académicas y de publicación considerando el volumen de uso, incluyen los siguientes formatos: Adobe Acrobat (.pdf), Adobe PostScript (.ps), Microsoft Word (.doc) y Microsoft PowerPoint (.ppt).

<sup>8</sup> Google Académico <http://scholar.google.com/>

Se plantea entonces la posibilidad de buscar un mecanismo que permita conservar estos enlaces externos, con el propósito de evitar la pérdida de la visibilidad Web del repositorio. En caso contrario, de no tomar medidas, miles de enlaces, indexaciones y referencias provenientes desde dentro y fuera del dominio ula.ve se perderán. Debido al gran volumen de enlaces, la alternativa de modificar manualmente los enlaces, portales externos e internos es prácticamente descartada y por tanto surge la idea de poder encausar automáticamente dichos enlaces externos hacia SABER-ULA con la nueva plataforma DSpace. Se debe poder garantizar que SABER-ULA migrado a DSpace pueda responder con los mismos contenidos a los cuales los enlaces externos hacían referencia. Como es de esperarse, al comparar los sistemas de repositorios institucionales, Alejandría (origen) y DSpace (destino), se pueden presentar diferencias entre el acceso e identificación de cada registro de información, los sistemas de consultas, estructuras de los enlaces y la plataforma empleada. Por tanto, se plantea la necesidad de un “conversor” que permita traducir o buscar los accesos y consultas equivalentes de uno a otro sistema y además adecuarlos al sistema destino. Tal como se puede observar en la figura 5, se pretende que un componente de software configurable, que sustituirá a Alejandría, convierta las solicitudes de información hacia el servicio de SABER-ULA en una respuesta automática de la plataforma DSpace. La idea es entonces establecer la relación entre ambos sistemas para que una vez realizada la migración sea transparente el acceso al RI.

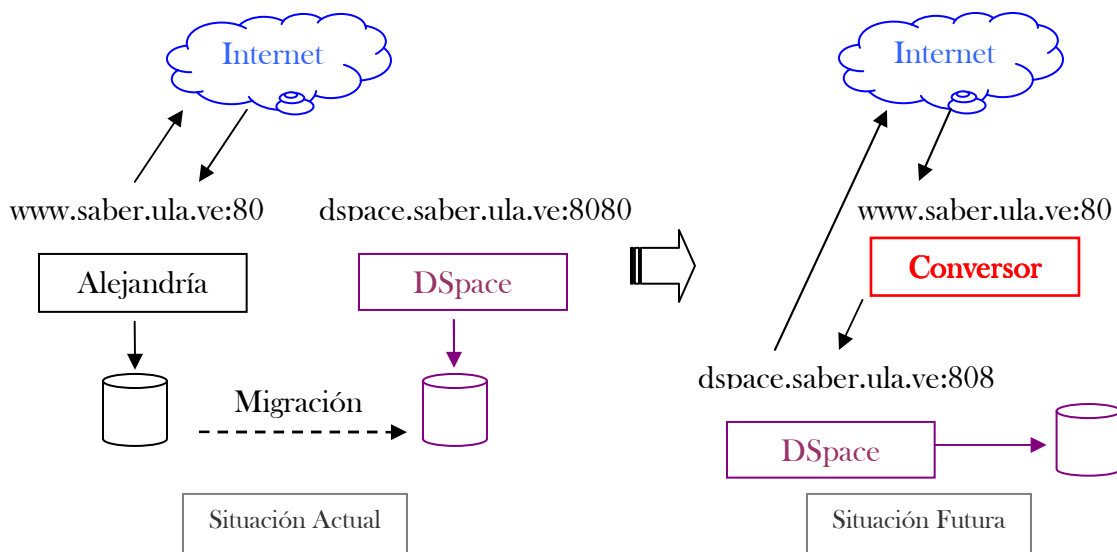


Figura 5. Modelo del problema de acceso automático de los requerimientos al servidor SABER-ULA a través del “conversor” luego de la migración.

El objetivo de este proyecto es desarrollar un sistema que permita mantener el acceso al servidor de SABER-ULA aunque cambie de forma, pues sus contenidos seguirán existiendo

aunque la manera de llegar a ellos cambie. Se pretende desarrollar una solución técnica, factible y general que permita mantener los enlaces actuales; accesos a documentos y consultas sobre la nueva plataforma sin cambiarlos, de manera que no se vea afectada la visibilidad, funcionalidad y uso actual del repositorio institucional, así como el prestigio que le aporta a la Universidad de Los Andes.

## **1.3 Objetivos**

### **1.3.1 Objetivo General**

Implementar un conversor de consultas y accesos a documentos desde el sistema automatizado de repositorio institucional Alejandría hacia DSpace, conservando la visibilidad Web del repositorio SABER-ULA.

### **1.3.2 Objetivos Específicos**

- Analizar los sistemas de consulta e identificación de documentos de ambos sistemas de repositorio institucional.
- Desarrollar un diseño general para la implementación de un conversor de consultas y accesos a documentos entre repositorios institucionales.
- Mantener la visibilidad Web del servidor SABER-ULA de forma transparente al usuario.
- Instalar y probar el conversor para el caso de las plataformas Alejandría y DSpace.
- Desarrollar un manual de usuario destinado a facilitar la instalación, la configuración, el mantenimiento y la posible modificación del sistema conversor.

## **1.4 Metodología**

### **1.4.1 Primera Fase: Investigación**

Esta fase pretende contextualizar el problema planteado y observar la importancia de su solución. Además, permite conocer los antecedentes y generar el marco teórico del proyecto.

#### **1.4.1.1 Revisión de conceptos:**

- Repositorio Institucional



- Metadatos
- DOI (Digital Object Identifier)<sup>9</sup>
- DSpace
- Alejandría
- Otros (servidor Web, HTTP, Handle System)

#### 1.4.1.2 Revisión de estadísticas:

- Revisión de las estadísticas de visibilidad y referencias hacia el servidor SABER-ULA
- Revisión de antecedentes
- Revisión de casos de migración de plataformas y el problema de mantener las consultas y accesos
- Revisión del plan de migración de la ULA

## 1.4.2 Segunda Fase: Análisis, diseño e implementación

Esta fase comprende las actividades de investigación, análisis del problema, diseño e implementación del conversor de accesos entre Repositorios Institucionales, lo que permite obtener la solución al problema planteado.

### *1.4.2.1. Análisis de los sistemas de consultas y direccionamiento de documentos:*

- DSpace (DOI, Handle System)
- Alejandría (Propietario)

### *1.4.2.2. Búsqueda de herramientas para redireccionar solicitudes Web:*

- Módulo mod\_rewrite de Apache
- PHP
- Otras (JavaScript, HTML)

### *1.4.2.3. Aplicación de pruebas básicas de redireccionamiento:*

Probar los casos específicos de consultas y accesos de Alejandría, con el fin de clasificar estos casos e identificar los mecanismos de redireccionamiento posibles para cada caso.

### *1.4.2.4. Diseño del diagrama general de la solución al problema: corresponde al diseño del componente de software y su funcionamiento:*

- Identificación de componentes
- Identificación de la relación entre los componentes y su funcionamiento.

---

<sup>9</sup> DOI (Digital Object Identifier) <http://www.doi.org/>

*1.4.2.5. Implementación del conversor: utilizando la metodología de enfoque evolutivo denominada “modelo de versiones” aplicable en sistemas pequeños y con ensamblaje de componentes (Sommerville, 2002):*

- Implementación de los programas de interpretación de la configuración del conversor
- Implementación de los programas de conversión de accesos, aplicado al problema específico
- Implementación de los programas de conversión de consultas, aplicado al problema específico

### **1.4.3 Tercera Fase: Instalación y pruebas**

En esta fase se instala el producto resultante de la fase anterior y se realizan las pruebas necesarias para garantizar su funcionamiento.

*1.4.3.1 Instalación final del conversor.*

*1.4.3.2 Determinación de los casos de prueba con datos migrados y no migrados para cada tipo de consulta y acceso.*

*1.4.3.3 Pruebas del conversor en un escenario simulado de migración.*

*1.4.3.4 Documentación sobre el proceso de instalación y configuración de los componentes del conversor (manual de administración).*

*1.4.3.5 Presentación de la solución al equipo de Migración del RI SABER-ULA para mostrar el funcionamiento.*

### **1.4.4 Cuarta Fase: Documentación final de Tesis**

En esta fase se pretende documentar en paralelo con otras fases el producto de la investigación teórica, el desarrollo, así como también los resultados y conclusiones de la tesis, entre otros.

*1.4.4.1 Documentación de antecedentes y marco teórico.*

*1.4.4.2 Documentación del análisis y diseño.*

*1.4.4.3 Documentación de la implementación y las pruebas.*

*1.4.4.4 Conclusiones y recomendaciones.*

## Capítulo 2 Marco Teórico

Este capítulo presenta el contexto que enmarca y sobre el cual se desarrolla este proyecto. Dicho contexto se inicia a raíz de un nuevo paradigma sobre el acceso y los derechos que los autores y los generadores de conocimiento tienen sobre sus productos, lo cual ha impulsado un movimiento ideológico y técnico que se denomina como Acceso Abierto al Conocimiento. Luego se detallan los diversos aspectos tecnológicos que hacen posible esta iniciativa destacando aquellos que serán utilizados en el desarrollo futuro de este proyecto. Para finalizar, se estudia el redireccionamiento de URL así como las diferentes herramientas que lo hacen posible.

### 2.1 Iniciativas de Acceso Abierto al Conocimiento

Según Núñez (2001, 2002) desde la década de los 80 surge una nueva forma de difundir la información gracias a Internet, pero las editoriales al darse cuenta de la capacidad de publicar y difundir en forma mas directa la información que de manera impresa, con costos cada vez menores y un alcance casi omnipresente, rápidamente desarrolla nuevos proyectos de apropiación del patrimonio intelectual de la comunidad académica. Así que se desarrolla una red de información paga y controlada, que coloca la información en las manos de cada investigador que posee los recursos para acceder a ella. Durante años se mantiene este modelo de publicación hasta que en los años 90 comienza a suceder una metamorfosis editorial

Una referencia importante de estos cambios en la publicación sucede en 1991 cuando, el investigador de física de Altas Energías del Laboratorio Nacional de los Álamos en Nuevo México EE.UU., Paul Ginsparg crea el Archivo de Borradores de física de Altas Energías<sup>10</sup>, como una iniciativa para distribuir electrónicamente obras no concluidas, pero bien estructuradas, entre colegas de un mismo campo para someterlas a discusión y luego publicarlas. Este sistema recibe hoy cerca de 30.000 borradores mensuales no sólo relacionados con la Física, como se había pensado en un principio, sino también con áreas como las matemáticas y las ciencias de la computación. Más de 160.000 usuarios acceden al archivo en todo el mundo,

---

<sup>10</sup> Sitio Web del Archivo de Borradores (preprint) de Física de Altas Energías <http://xxx.lanl.gov/>

teniendo acceso al conocimiento casi instantáneamente, además con un formato bien definido y citas entrelazadas en una misma base de datos.

### 2.1.1 Iniciativa de Acceso Abierto de Budapest

La Iniciativa de Acceso Abierto de Budapest (BOAI)<sup>11</sup>, surge en Budapest a raíz de una reunión convocada por el instituto para una sociedad abierta (OSI, Open Society Insitute) en diciembre del 2001, con el fin de acelerar el proceso mediante el cual los artículos de investigación en cualquier área académica estuvieran disponibles de forma gratuita en Internet. Los participantes exploraron cómo las iniciativas por separado podían llegar, trabajando juntas al éxito. Exploraron las estrategias para servir a los intereses de la investigación, los investigadores y de las instituciones y sociedades que mantienen la investigación. Y finalmente exploraron como el OSI y otras fundaciones podían usar sus recursos de una manera más productiva para ayudar a la transición al acceso abierto y hacer la publicación económicamente auto-sustentable.

Según la BOAI (Soros, 2002), para lograr el acceso abierto a la literatura periódica académica, se recomienda:

- *Auto-Archivado*: Consiste en proveer a los académicos de herramientas para depositar sus artículos en archivos electrónicos abiertos, de tal forma que alcanzando los estándares creados por la OAI<sup>12</sup>, las herramientas como los meta-busadores pueden centralizar el acceso a diferentes instancias de los archivos.
- *Publicaciones periódicas de acceso abierto*: Los académicos necesitan los medios para crear una nueva generación de publicaciones periódicas comprometidas con el acceso abierto y para ayudar a las ya existentes para hacer la transición al acceso abierto. Estas publicaciones no podrán invocar restricciones de acceso por asuntos de *copyright* del material que publican, sino usar el *copyright* y otras herramientas para asegurarse del permanente acceso abierto a todos los artículos que publiquen. Los gastos de publicación no deben ser cubiertos por cuotas de suscripción o acceso, existen muchas otras fuentes de financiamiento tales como fundaciones y financiamiento gubernamental, universitario, o por parte de laboratorios de investigación.

---

<sup>11</sup> Sitio Web de la Iniciativa de Budapest para el Acceso Abierto: <http://www.soros.org/openaccess>

<sup>12</sup> OAI: Open Access Initiative (<http://www.openarchives.org/>)

## 2.1.2 Declaración de Berlín sobre Acceso Abierto al Conocimiento

Esta declaración fue redactada para promover Internet como el instrumento funcional que sirva de base global del conocimiento científico y la reflexión humana, y para especificar medidas que deben ser tomadas en cuenta por los encargados de las políticas de investigación, y por las instituciones científicas, agencias de financiamiento, bibliotecas, archivos y museos<sup>13</sup>.

Según esta declaración (Sociedad Max Planck, 2003), la meta es apoyar nuevas posibilidades de diseminación del conocimiento, no solo a través de la manera clásica, sino también utilizando el paradigma del acceso abierto por medio de la Internet. De manera que los productores de conocimiento científico deben estar comprometidos a establecer el acceso abierto como un procedimiento meritorio, y contribuir con dicho paradigma por medio de resultados de investigaciones científicas, datos primarios y metadatos, materiales fuentes, representaciones digitales de materiales gráficos y pictóricos, y materiales eruditos en multimedia, satisfaciendo las siguientes condiciones:

- El autor y depositario de la propiedad intelectual debe garantizar a todos los usuarios por igual, el derecho gratuito, irrevocable y mundial de acceder a un trabajo erudito. Licencia para copiarlo, usarlo, distribuirlo, transmitirlo y exhibirlo públicamente, y para hacer y distribuir trabajos derivados, en cualquier medio digital para cualquier propósito responsable, sujeto al reconocimiento apropiado de la autoría.
- Se debe depositar una versión completa del trabajo y todos sus materiales complementarios en por lo menos un repositorio online, que utilice estándares técnicos aceptables, que sea apoyado y mantenido por una institución académica, sociedad erudita, agencia gubernamental, o una bien establecida organización que busque implementar el acceso abierto, distribución irrestricta, interoperabilidad y capacidad archivística a largo plazo.

Esta declaración de Berlín de acceso libre al conocimiento de las ciencias y las humanidades reúne actualmente a 249 organizaciones y universidades alrededor del mundo<sup>14</sup>, significando un hito importante en la consolidación del movimiento institucional del acceso abierto al conocimiento. Estas iniciativas se complementan y apoyan en desarrollos de técnicas de intercambio de información y uso de estándares de diferente tipo en los servidores de publicación. En la siguiente sección se mencionan los estándares más difundidos.

---

<sup>13</sup> Berlín Declaration (Declaración de Berlín): <http://www.zim.mpg.de/openaccess-berlin/berlindeclaration.html>

<sup>14</sup> La Universidad de Los Andes, Mérida Venezuela se incorpora a la declaración de Berlín en el año 2006, tal como puede verificarse en <http://oa.mpg.de/openaccess-berlin/signatories.html>

## 2.2 Tecnologías de Acceso Abierto

En esta sección, se presentan las tecnologías y estándares del acceso abierto organizadas según su uso y aplicación en las diferentes fases de la publicación. Se comienzan con la publicación en sí, es decir la forma en que se visualiza y expresa la información, refiriéndonos a estándares de interfaz y contenido. Luego continuamos con los mecanismos necesarios para la identificación de esos contenidos en Internet y se finaliza con la tendencia actual de publicación, difusión e intercambio de contenidos conocido como Repositorios Institucionales. Finalmente, se mencionan detalles técnicos sobre las plataformas de los casos de estudio de esta propuesta.

### 2.2.1 Estándares de Interfaz y Contenido

#### World Wide Web

La Web es el estándar más importante para publicar texto e hipermedia, el cual esta caracterizado por ser una tecnología simple, escalable y eficiente. Desde su origen (Berners-Lee, 1990) ha resultado de amplia utilidad para el formato de la información, relacionando recursos y medios. Su definición contempla desde la identificación y representación de los recursos, hasta los protocolos de intercambio a través de la red.

El Servidor Web, permite implementar el *protocolo HTTP (Hypertext Transfer Protocol)*, el cual está diseñado para transferir hipertextos, páginas Web o páginas HTML (hypertext markup language): textos complejos con enlaces, figuras, formularios, botones y objetos incrustados como animaciones o reproductores de música. Un Servidor Web se mantiene a la espera de *peticiones HTTP* realizadas por un *cliente HTTP* que generalmente denominamos *navegador*. El navegador realiza una petición al servidor y éste le responde con el contenido que el cliente solicita.

El protocolo HTTP, o protocolo de transferencia de hipertexto es el protocolo usado en cada transacción de la Web. El HTTP fue desarrollador por el consorcio W3C<sup>15</sup> y la IETF<sup>16</sup>. HTTP sigue el esquema petición-respuesta entre un cliente y un servidor. Al cliente que efectúa la petición (un navegador) se lo conoce como "user agent" (agente del usuario) y a la información transmitida se la llama recurso y se la identifica mediante un URL. Los recursos pueden ser:

---

<sup>15</sup> Consortium World Wide Web (W3C) <http://www.w3.org/>

<sup>16</sup> Internet Engineering Task Force (IETF) <http://www.ietf.org/>

archivos, el resultado de la ejecución de un programa, una consulta a una base de datos, la traducción automática de un documento, etc. (Spainhour y Eckstein, 1999).

Uno de los servidores mas conocidos, libre y de código abierto, que implementan el protocolo HTTP es el Servidor Apache<sup>17</sup>; un software para plataformas Unix, Windows, Macintosh y otras, que específicamente implementa el protocolo HTTP/1.1 y la noción de sitio virtual. Apache es desarrollado dentro del proyecto HTTP Server (httpd) de la Apache Software Foundation<sup>18</sup> y presenta entre otras características mensajes de error altamente configurables, bases de datos de autenticación y negociado de contenido (Laurie, 2002).

### **Métodos de envío de peticiones**

Un servidor Web recibe las peticiones enviadas desde un cliente a través de uno de los métodos de envío del protocolo HTTP, entre estos métodos tenemos: OPTION, GET, HEAD, DELETE, OPEN y POST. Los métodos que serán tomados en cuenta en el proyecto serán los que implican consultas y retorno de información o archivos con contenidos (GET y POST), y se describen a continuación.

El método de envío de peticiones GET se utiliza para recuperar información identificada por un URI<sup>19</sup>. Generalmente se utiliza solo para consultas al servidor, pero también puede ser utilizado para cualquier otro tipo de petición. Los datos a enviar son insertados en el URL especificado, implicando que pueden ser vistos en la barra de direcciones del navegador cuando se realiza la solicitud. Uno de sus usos mas frecuentes es la transferencia de una pequeña cantidad de información (utilizada para realizar consultas a una base de datos) al servidor en forma de pares atributo-valor añadidos al final del URL, luego del caracter “?” (W3C, 1994). La longitud de las peticiones GET está limitada por el espacio libre en los buffers de entrada, por lo que no se debe utilizar para enviar gran cantidad de información.

El método POST realiza una transacción mediante el protocolo HTTP, que se encarga de enviar información incluida en el cuerpo de la petición para que sea procesada por el recurso identificado, por esto, los datos enviados no se ven en la barra de direcciones del navegador. La función que realiza el método es asignada por el servidor Web, generalmente mediante el URI de la petición, así, la entidad enviada es subordinada a ese URI. Este método es más fiable porque no utiliza el caracter “?” para dirigir los datos, sino que las envía a través de

---

<sup>17</sup> Apache HTTP Server Project: <http://httpd.apache.org/>

<sup>18</sup> Apache Software Foundation: <http://www.apache.org/>

<sup>19</sup> Uniform Resource Identifier <http://www.ietf.org/rfc/rfc2396.txt>

la entrada estándar, lo que significa virtualmente poder recibir cualquier tamaño de datos (W3C, 1994).

## **Metadatos**

Según las investigaciones de Torrén (2003), los metadatos han sido usados principalmente en conexión con datos discretos como datos alfanuméricos y numéricos almacenados en una base de datos convencional. Además, Torrén menciona que los metadatos pueden ser usados para dar información acerca del contenido de un recurso de información, con el objetivo de facilitar el "descubrimiento", identificación, localización, detección preventiva de su uso ilegítimo y generar a partir de este tipo de datos definiciones conocidas como "diccionarios de datos".

Se puede definir a los metadatos como datos acerca de datos, es decir como información descriptiva usada para consultar. Algunos metadatos pueden ser realizados mecánicamente, tales como tamaños de archivos, sumas e indexaciones de texto completo. Otros metadatos son considerados como un orden superior de las descripciones humanas tales como títulos, autores, identificadores únicos y resúmenes.

Con el uso frecuente y creciente del Web para la publicación comenzó a hacerse necesario la incorporación de metadatos como elemento adicional a los contenidos, con el fin de facilitar la descripción, uso y recuperación de información. Justamente, la iniciativa de metadatos Dublín Core<sup>20</sup> (DCMI por sus siglas en inglés) trata de fomentar la adopción a gran escala de estándares de metadatos interoperables y de promover el desarrollo de vocabularios especializados de metadatos para la descripción de recursos, soportando un amplio rango de propósitos y modelos. Sus actividades incluyen el trabajo en la arquitectura y modelado, discusiones y trabajo colaborativo en las comunidades DCMI y sus grupos de expertos, conferencias anuales, coordinación de los estándares y realizar esfuerzos educativos para promover la aceptación global de los estándares de metadatos y sus prácticas (DCMI, 2006).

Su principal producto es el modelo de metadatos llamado Dublín Core. El nombre proviene de Dublín (Ohio, USA), allí se albergó en 1995 la primera reunión de especialistas en metadatos a nivel mundial. Este modelo consta de 15 definiciones semánticas descriptivas que se muestran en la figura 6, estas definiciones pretenden transmitir un significado semántico a las mismas. El modelo se diseñó para proporcionar un vocabulario de características básicas, capaz de proporcionar la información descriptiva básica sobre cualquier recurso, sin que importe el formato de origen, el área de especialización o el origen cultural (Ayers, et al., 2001).

---

<sup>20</sup> Dublin Core <http://dublincore.org>



<b>Elementos de Dublín Core</b>
1. título (title),
2. asunto (subject),
3. descripción (description),
4. fuente (source),
5. idioma (language),
6. relación (relation),
7. cobertura (coverage),
8. creador (creator),
9. editor (publisher),
10. contribuidor (contributor),
11. derechos (rights),
12. fecha (date),
13. tipo (type),
14. formato (format),
15. identificador (identifier).

Figura 6. Listado de los elementos del Dublín Core. Tomado de <http://dublincore.org>

Estos elementos pueden ser clasificados en tres grupos que indican la clase o el ámbito de la información almacenada:

- Elementos relacionados con el contenido del recurso (título, asunto, descripción, fuente, idioma, relación, cobertura).
- Elementos relacionados con el recurso cuando es visto como una propiedad intelectual (creador, editor, contribuidor, derechos).
- Elementos relacionados con la instalación del recurso (fecha, tipo, formato, identificador).

De esta manera, los metadatos de Dublín Core, por su generalidad, se utilizan principalmente para describir los recursos de cualquier sistema de información y por eso están siendo utilizados como la base para muchos sistemas descriptivos, como por ejemplo: organizaciones educativas, bibliotecas, instituciones del gobierno, sector científico de la investigación, páginas Web, negocios que requieren lugares más investigables, entre otros.

## 2.2.2 Estándares o Mecanismos de Identificación de Documentos

La necesidad de identificar objetos tiene muchos ejemplos en la historia de la computación. Un de ellos es la conocida noción de “espacios de nombres” (*namespaces*), que actualmente es una parte fundamental de la infraestructura del Web a través de la identificación única de sus elementos o recursos. Los “espacios de nombres” en la Web, son dominios de nombres asociados con elementos que junto con el nombre del elemento produce un URL que

identifica unívocamente el elemento. Dentro de las iniciativas y las tecnologías de acceso abierto al conocimiento esta necesidad aumenta, pues la identificación del contenido publicado requiere ser referenciada de manera clara, única y sin ambigüedad de la misma forma que en la publicación tradicional<sup>21</sup>.

### **Identificadores de Objetos Digitales**

El sistema DOI<sup>22</sup> se utiliza para identificar objetos contenidos en el ambiente digital. Los identificadores DOI son asignados a cualquier entidad para su uso en redes digitales y son usados para proveer información general y donde pueden ser encontrados en Internet. La idea es que, la información sobre un objeto digital puede cambiar a través del tiempo, incluso su ubicación, pero su identificador DOI no cambiará.

El sistema DOI provee una estructura para la identificación persistente, manejando contenido intelectual, manejando metadatos, relacionando los clientes con proveedores de contenido, facilitando el comercio electrónico y permitiendo el manejo automatizado de los medios de comunicación. Los identificadores DOI pueden ser usados para cualquier tipo de manejo de datos, sea comercial o no (DOI, 2006). Además, el sistema es gerenciado por la “Internacional DOI Foundation”, un consorcio de membresía abierta que incluye a patrocinadores comerciales y no comerciales, y recientemente ha sido aceptado para la estandarización ISO. Han sido asignados hasta noviembre del 2007 mas de 28 millones de identificadores DOI (MEDRA, 2004).

El sistema DOI utiliza al Handle System, con el cual se permite la identificación persistente y semánticamente interoperable de las entidades de propiedad intelectual (Handle.net, 2005). El Handle System provee un servicio de identificación global de propósito general permitiendo una identificación segura en Internet, el cual fue diseñado para permitir que una amplia colección de comunidades use la tecnología para identificar contenido digital independiente de su localización.

El Handle System esta implementado como un sistema de información distribuido diseñado para proveer un eficiente extensible y seguro servicio global de nombre para ser usados en Internet. Este protocolo abierto permite almacenar nombres, o “handles”, de recursos digitales y resolverlos para localizar, acceder y hacer uso del recurso. Este sistema,

---

<sup>21</sup> Por ejemplo con identificadores como el ISBN (International Standard Book Number)

<sup>22</sup> DOI (Digital Object Identifier) <http://www.doi.org/>

especificado como estándar de la IETF, es la base de servicios importantes como DNS<sup>23</sup>, LDAP<sup>24</sup> y URN<sup>25</sup> (Sun, 2003).

### 2.2.3 Herramientas de Publicación Electrónica

La publicación electrónica es una recomendación importante de las iniciativas de acceso abierto al conocimiento en todos los ámbitos. Esta puede llevarse a cabo de muchas maneras y con muchas aplicaciones, pero paralelamente al movimiento OAI han surgido desarrollos abiertos de diversas plataformas para que las instituciones productoras de conocimiento no tengan dificultad para lograr los objetivos de la Declaración de Berlín. Inicialmente, se presenta en esta sección a los Repositorios Institucionales como la materialización de la iniciativa de preservación y difusión de conocimientos. Luego se mencionan aspectos esenciales de las plataformas de estudio (Alejandría y DSpace) y sobre el servicio de publicación de acceso abierto de la Universidad de Los Andes, denominado SABER-ULA, considerado en particular para este proyecto.

#### ¿Que es un Repositorio Institucional (RI)?

Es un conjunto de servicios de almacenamiento, gestión y disseminación de materiales digitales disponibles a los miembros de una determinada comunidad o institución. La importancia de un RI, se basa en los contenidos del repositorio, la calidad, la constante actualización, la seguridad, la facilidad de acceso a sus contenidos y la amplitud de su difusión. Adicionalmente, el desarrollo de un RI genera una memoria documental de conocimientos experiencias y productos del quehacer de cada institución, y la posibilidad de disponer en Web de este tipo de memoria permite avanzar en la construcción y consolidación de una comunidad científica al no tener que empezar desde cero (Dávila et al, 2005).

El término Repositorio Institucional implica que el sistema está diseñado para soportar y archivar las necesidades y metas de la institución, mas no necesariamente las del individuo. Sugiere que las contribuciones de materiales al repositorio sirven para resaltar los logros de la institución, en vez de aquellos provenientes de investigaciones individuales. El contexto más adecuado para hablar de un RI es el universitario, en donde éste debe proveer un lugar para el

---

<sup>23</sup> *Domain Name Service* [www.dns.net](http://www.dns.net)

<sup>24</sup> *Lightweight Directory Access Protocol* <http://www.openldap.org/>

<sup>25</sup> *Uniform Resource Name* <http://www.urn-info.org/>

profesorado, tesis, diarios electrónicos, entre otros. Los repositorios institucionales proveen a la institución un mecanismo para exhibir su potencia académica, centralizar e introducir eficiencia a la administración de documentos digitales de valor, y responder proactivamente a la creciente crisis de su comunidad. Fried y Gibbons (2005), menciona que la disponibilidad de plataformas para implementar repositorios institucionales de acceso abierto ha promovido una proliferación de RRII a nivel mundial, particularmente entre las instituciones académicas y de investigación.

### **El Repositorio Institucional (RI) SABER-ULA.**

SABER-ULA<sup>26</sup> es una iniciativa de la Universidad de Los Andes (ULA) destinada a preservar el patrimonio intelectual universitario y a constituirse en un instrumento de difusión de las capacidades y potencialidades del personal docente y de investigación de la ULA. Además, según Dávila y colaboradores (2005) esta iniciativa pudiera servir como un mecanismo que posibilite su auditabilidad social considerando que para una comunidad académica es vital conservar y difundir su patrimonio intelectual. Por lo tanto, se deben generar políticas, mecanismos e incentivos que posibiliten preservarlo y difundirlo, permitiendo el acceso a ese conocimiento al mayor número de personas posible.

SABER-ULA consta de cuatro unidades fundamentales dentro de los contenidos que preserva y difunde:

- *Investigación y Postgrado:* Se encuentran reseñados cientos de unidades de investigación y recogidos los perfiles profesionales de un poco más de un millar de investigadores de la ULA. Contiene las fichas básicas de identificación de cada unidad de investigación e investigador con los datos mínimos de contacto y adscripción, enlace a su unidad y a sus publicaciones electrónicas en el repositorio.
- *Revistas Electrónicas:* Se encarga del proceso de apropiación tecnológica por parte de comunidades organizadas, además de difundir en formato electrónico las publicaciones académicas impresas de la ULA y fomentar la edición de versiones solo digitales.
- *Servidores Temáticos:* Generan y concentran información catalogada o certificada de áreas de conocimiento y mantiene mecanismos de interacción para la comunidad de información de esas áreas.

---

<sup>26</sup> SABER-ULA <http://www.saber.ula.ve>

- *Cartelera de Eventos:* Disemina la información auto archivada de los eventos que organizan las distintas dependencias académicas de la institución.

Según señala Fried y Gibbons (2005), el desarrollo de un RI depende de la cultura y bienes de la institución, por lo tanto cada una de ellas tiene su propio proceso para conseguir su RI. Efectivamente el desarrollo de SABER-ULA surgió con la necesidad de preservación, de centralización de la información de los entes que producen conocimientos (unidades de investigación y postgrados) y como una alternativa a los altos costos de publicación en papel para sus revistas. Luego amplía su alcance a la publicación de otros documentos no arbitrados incorporándose a la tendencia mundial del acceso abierto.

### **Plataformas para la Publicación Electrónica**

Entre los sistemas de tipo privativo para la publicación electrónica se encuentra Alejandría, la cual es una plataforma que facilita el desarrollo y la implantación de sistemas de teleinformación considerando la arquitectura de la información. Según sus creadores (Alejandría, 2006), Alejandría nace de la fusión del conocimiento de las ciencias de la información (bibliotecología, archivología, gestión documental, redes de información, etc.) con el conocimiento de tecnologías de la información (sistemas, software, bases de datos, informática, computación, tecnologías Web, etc.).

El servidor de Alejandría trabaja sobre las plataformas Windows (NT, 2000, 2003), Usa una base de información que se implanta sobre plataformas de base de datos relacionales estándares y una arquitectura de múltiples capas que permite la estandarización en el manejo de las transacciones propias de un sistema bibliotecario.

El manejador de información realiza automáticamente las siguientes operaciones:

- Manejo automático de operaciones de alto nivel sobre objetos de información, cual permite un modelado de objetos de información personalizables.
- Generación automática de estadísticas de módulos, vistas, objetos de información y servicios. Se manejan varios tipos de documentos permitiendo la apertura automática de los correspondientes visores.
- Procesos automáticos de conversión de datos.
- Seguridad incorporada al manejador de información.
- Manejo de transacciones de información en tiempo real, debido a que incluye el manejo de actualización de índices en tiempo real y manejo de índices por tablas invertidas en línea.

Por otra parte, también existen otras plataformas para publicación electrónica de tipo abierto o libre (Wyles, R. et al, 2006), una de sus mayores exponentes es la plataforma DSpace<sup>27</sup>. DSpace, es un paquete de software libre que provee las herramientas para el manejo de recursos digitales, comúnmente usado como la base para un Repositorio Institucional (RI). También se conoce como una plataforma para actividades de preservación digital. Desde su salida en el 2002, como un producto de la alianza HP-MIT, (hasta enero del 2008) ha sido instalado y esta en producción en mas de 315 instituciones, de 52 ciudades alrededor del mundo.

Técnicamente, DSpace esta basado en estándares y plataformas de software libre. Es compartido bajo licencia BSD, la cual permite a los usuarios personalizarlo o extenderlo de acuerdo a las necesidades. Esta escrito en Java y JSP, usando el Java Servlet Framework. Usa una base de datos relacional, soportando el uso particular de PostgreSQL y Oracle. Este sistema emplea principalmente una interfaz Web y también soporta el protocolo de recolección OAI-PMH<sup>28</sup> v2.0 exportando también paquetes METS (Metadata Encoding and Transmission Standard). Además, DSpace utiliza una versión de metadatos desarrollada por la DCMI llamada Dublín Core a través de todo el contenido (DSpace, 2007).

DSpace soporta una variedad considerable de datos, incluyendo libros, tesis, fotos, películas, videos, datos de investigación y otros. Además acepta cualquier tipo de formato digital. Algunos ejemplos de datos que DSpace puede contener son (DSpace, 2007):

- Documentos, tales como artículos, borradores, reportes técnicos, documentos de conferencias.
- Libros publicados, tesis.
- Conjuntos de datos, registros administrativos, diarios.
- Programas de computadoras, visualizaciones, simulaciones y modelos, objetos de aprendizaje, páginas Web
- Publicaciones multimedia, imágenes, archivos de audio, archivos de video, colecciones de bibliotecas digitales reformateadas.
- Conjunto de datos bibliográficos

---

<sup>27</sup> DSpace <http://www.dspace.org>

<sup>28</sup> *Open Archives Initiative Protocol for Metadata Harvesting* <http://www.openarchives.org/pmh/>

## 2.3 Herramientas de redireccionamiento de URL

En esta sección se describen algunas de las herramientas usadas para realizar redireccionamiento o reescritura de URL, Teóricamente, en un mundo perfecto el redireccionamiento no es necesario, pero en la práctica es muy común cambiar archivos de sitio dentro de un servidor Web (o incluso a un servidor distinto), y es muy común darse cuenta que no todas las cosas se encuentran en el sitio deseado ni en el momento adecuado (Laurie, 2002). El redireccionamiento permite que luego de haber movido documentos, realizado cambios o de haber reconfigurado un servidor Web, las peticiones realizadas por un cliente puedan ser distribuidas adecuadamente a través del sistema de archivos del servidor, o a través de la Web, sin necesidad de cambiar los enlaces existentes en los archivos HTML.

Este proyecto de grado se desarrolla tomando como punto inicial la migración de la plataforma sobre la que se soporta un repositorio institucional, esta migración provoca cambios al servidor Web como los descritos anteriormente. La solución a la pérdida de la visibilidad Web a raíz de la migración; es utilizar una herramienta de redireccionamiento de las peticiones realizadas. La sección presenta diversas opciones de redireccionamiento de las cuales se pretende escoger la(s) que se adapte(n) mejor al caso de estudio.

### Apache Alias

Este módulo perteneciente a la familia Apache, es uno de los mas utilizados en los casos de almacenamiento de documentos fuera del directorio raíz del servidor, y en los casos en los que es necesario mover directorios completos dentro del mismo, permitiendo el redireccionamiento automático de todos los archivos pertenecientes al directorio movido dentro del servidor. A continuación se mencionan las directivas mas usadas del módulo Alias:

1. Script: Esta directiva agrega una acción que activa un cgi-script cuando se solicita un archivo utilizando un método específico. Envía el URL y la ruta al archivo del documento solicitado utilizando variables ambientales estándar CGI<sup>29</sup> como; PATH\_INFO y PATH\_TRANSLATED.
2. ScriptAlias: Es una directiva que permite que los scripts sean almacenados de manera segura fuera del alcance de hackers y crackers, y adicionalmente marca el directorio donde son almacenados como contenedor de scripts CGI.

---

<sup>29</sup> Common Gateway Interface <http://www.w3.org/CGI/>

3. **ScriptAliasMatch:** Utilizada para emparejar un URL solicitado con una expresión regular dada, en este caso el servidor sustituye cada emparejamiento proveniente del URL original y la utiliza como un nombre de archivo añadiéndolo al URL final.
4. **Alias:** Esta directiva es utilizada para mapear un URL recurso a su dirección física en el sistema de archivos.
5. **AliasMatch:** Esta directiva actúa del mismo modo que la anterior, solo que se activa al emparejar el URL original con una expresión regular.
6. **UserDir:** En este caso el servidor redirecciona una solicitud de un directorio local que se encuentra en algún lugar de computador cuyo DNS empareja el solicitado por el cliente, hacia su directorio real dentro del sistema de archivos.
7. **Redirect:** La directiva **Redirect** mapea un URL antiguo hacia uno nuevo. El cliente recibe el nuevo URL, el cual solicita una vez más la información a la nueva dirección.

El siguiente ejemplo muestra el uso de la directiva **Alias**:

```
...  
Alias /<otro sitio>/ /<ruta al sitio>/<otro sitio>  
...
```

### **Apache mod\_rewrite**

El `mod_rewrite` es uno de los módulos de redireccionamiento mas populares del servidor Web Apache, pero del mismo modo es uno de los mas complejos de comprender y utilizar (Bohen, 2006). Este módulo es una máquina de reescritura de URL basada en reglas conformadas por comandos y expresiones regulares. Provee un mecanismo de manipulación de URL muy poderoso que depende de una diversidad de pruebas, variables del servidor, variables ambientales, cabeceras HTTP, etc. Incluso provee la herramienta para conectarse a bases de datos externas al servidor de manera de comparar total o parcialmente un URL para su posterior reescritura (apache.org, 1996).

El módulo opera sobre el URL completo, incluyendo la parte de la información de las rutas, tanto en el contexto interno del servidor (archivo `httpd.conf`) como en el contexto de directorios (archivo `.htaccess`). Este módulo puede realizar casi cualquier tipo de manipulación, sobre cualquier tipo de URL (apache.org, 1997), el administrador de un sitio Web lo puede utilizar en cuatro tipos de situaciones:



- Dentro del archivo de configuración del servidor (httpd.conf) para ser aplicado en cualquier contexto. Las reglas se aplican a todos los URL del servidor principal y del virtual.
- Dentro de cada servidor virtual. En este caso las reglas de redireccionamiento se aplican solo a los URL del servidor virtual.
- Dentro de cada directorio. Las reglas se aplican solo al directorio especificado.
- Dentro del archivo “.htaccess” del servidor. En este caso los usuarios también puedes utilizar el módulo para el directorio especificado.

El siguiente ejemplo muestra el uso de algunas de las directivas del módulo de redireccionamiento:

```
RewriteEngine On
RewriteCond %{Request_URI} !^/index.html
RewriteMap tolower int:tolower
RewriteRule ^/articles/(.*)\.html /articles/${tolower:$1}.html [R]
```

En el ejemplo mostrado, luego que se enciende el módulo de redireccionamiento; existe una regla condicional que verifica si en el URI solicitado existe la frase “index.html”. En caso de que la condición sea positiva, se especifica que se utilizará un mapa de funciones internas del módulo que se encarga de llevar el texto resultante a letras minúsculas. Finalmente, la última línea del ejemplo indica la expresión regular que debe evaluarse para aplicar el mapa de redireccionamiento.

El conjunto completo de directivas de redireccionamiento de mod\_rewrite se especifica detalladamente en el anexo A1.

### **Etiqueta Meta Refresh**

Las etiquetas Meta son etiquetas HTML que contienen información acerca de una página Web que no es desplegada directamente al cliente que la solicita. Generalmente encontramos en ellas: Palabras clave (name=“keywords”) utilizadas para aumentar la visibilidad a los buscadores Web, descripciones (name=“description”) que proveen una sinopsis corta de la página, definiciones de elementos como la fecha de expiración de la página (http-equiv=“expires”) y etiquetas que informan al navegador o al servidor Web que deben realizar alguna acción sobre la página (http-equiv=“refresh”) (Webdesign.com, 2001).

La etiqueta **Meta Refresh** es comúnmente utilizada para recargar y redireccionar páginas Web. Entre sus usos está la recarga de páginas con contenido dinámico, y el redireccionamiento hacia una nueva página (Webdesign.com 2002).

En el caso de la recarga de una misma página Web su sintaxis es de este modo:

```
<meta http-equiv="refresh" content="600">
```

Donde “meta” es la etiqueta perteneciente a la sección <head> o cabecera de un archivo HTML. “http-equiv=“refresh”” es un atributo que informa al navegador que su etiqueta “meta” esta enviando un comando HTTP y no una etiqueta estándar. “refresh” es un comando de cabecera HTTP utilizado para que el servidor Web sea notificado que la página va a ser recargada. El tiempo en segundos para que el navegador recargue la página viene dado por el atributo “content=“600””.

Mientras que en el caso del redireccionamiento a una página nueva viene dado de la siguiente manera:

```
<meta http-equiv="refresh" content="2;url=http://www.paginanueva.com">
```

Diferenciado de la recarga por el contenido del atributo “content”, ya que en este caso el número es el tiempo en segundos para que se produzca el redireccionamiento, y luego, separado por punto y coma (;), se encuentra el URL al cual la página Web va a ser redireccionada.

### **JavaScript window.location**

Los Objetos en JavaScript son colecciones de piezas de datos llamadas propiedades del objeto, a las cuales se les refiere por medio del objeto seguido de un punto, y a su vez la propiedad deseada (objeto.propiedad), estas propiedades pueden ser variables que contienen cualquier tipo de datos, incluyendo arreglos, funciones, y otros objetos. Un objeto también puede servir como arreglo asociativo, el cual es un tipo de dato muy poderoso para la programación. En este caso la sintaxis cambia: una cadena con el nombre de la propiedad deseada es encerrada por corchetes (objeto[“propiedad”]).

Cada una de las ventanas y marcos en un navegador Web está representada por un objeto “Window”, que es la raíz de su propia jerarquía y el objeto global (Mundogeeek, 2004). Entre las propiedades mas interesantes de este objeto encontramos una llamada “location”, que

es una referencia hacia el objeto “Location”, que a su vez es una representación del URL del documento mostrado en la ventana del navegador (Flanagan, 1997). Este objeto permite obtener o modificar la dirección Web en la que se encuentra el usuario, es comúnmente utilizado luego de un cambio de dominio para redireccionar a una nueva página Web; ya que además de contener el URL de la página Web, es capaz de controlarlo. Al asignarle una cadena que contenga un URL al objeto “Location”, el navegador responde cargando el URL recién especificado y desplegando en la ventana el documento al que se refiere.

La siguiente es la sintaxis utilizada para redireccionar un URL mediante el objeto “Location”:

```
<script type="text/javascript">
window.location = http://www.paginanueva.com/;
</script>
```

Tizag.com (2000) recomienda utilizar este tipo de redireccionamiento en el caso de cambiar el URL de un sitio Web, o al haber movido un archivo a un lugar diferente, mas no lo recomienda en el caso en que el redireccionamiento pueda ser reemplazado fácilmente con un simple enlace HTML.

### **PHP: header()**

La función **header()** se utiliza al comienzo de un fichero HTML para enviar las cadenas de texto de la cabecera HTTP. **header()** debe llamarse antes de que se genere salida alguna en el script que la usa, bien con etiquetas HTML normales o con PHP. Un error muy frecuente consiste en leer código con **include()** y que dicho código inserte espacios o líneas en blanco antes de llamar a **header()** (PHP.net, 2001).

Hay dos casos especiales de llamadas a “header”. La primera es la cabecera "Location". No sólo envía esta cabecera al navegador, sino que también devuelve un código de estado REDIRECT a Apache. Desde el punto de vista del programador de scripts esto no debería ser importante, pero para quien comprende las interioridades de Apache sí puede serlo.

```
header("Location: http://www.php.net"); /* Redirecciona el navegador al sitio Web
de PHP */
exit; /* Asegura que el código que le sigue no se ejecute luego del
redireccionamiento. */
```

El segundo caso especial se produce con cualquier cabecera que comience con la cadena, "HTTP/" (las mayúsculas no son significativas). Por ejemplo, si tiene la directiva `ErrorDocument 404` de Apache apuntando a un script PHP.

```
header("http/1.0 404 Not Found");
```

Todas las herramientas de redireccionamiento de URL mencionadas en la sección serán consideradas para la implementación del proyecto, de las cuales; como ya se mencionó, se utilizará(n) la(s) mas adecuada(s) para cada problema de cambio de redirección que se presente.

## Capítulo 3 Análisis

En este capítulo se presenta un análisis de los sistemas de consulta y direccionamiento de documentos en los sistemas de repositorios institucionales Alejandría y DSpace, haciendo énfasis en las características morfológicas de los URL, así como las del funcionamiento de cada uno de ellos. Finalmente se realiza una comparación de esas características entre ambos sistemas, tomando en cuenta sus similitudes y diferencias. También se pretende enlazar cada uno de los sistemas en estudio, de manera que se pueda obtener una idea general de la solución del problema.

### Morfología del URL

El URL (Uniform Resource Locator) puede ser dividido en primera instancia en dos partes: El nombre del esquema utilizado (<scheme>), y seguido de dos puntos (:), una cadena de caracteres que se interpreta dependiendo del esquema utilizado (<scheme-specific-part>), siendo su sintaxis muy sencilla:

<scheme>:<scheme-specific-part>

Un URL es una secuencia de caracteres que puede ser representado de distintas maneras, en la mayoría de los esquemas cada parte del URL posee una secuencia de caracteres que representa secuencias de bytes utilizadas en los protocolos de Internet. Cada byte puede ser representado gráficamente por el carácter que lo posee, y adicionalmente, en el caso en que los bytes no posean una representación gráfica, estos pueden ser representados por una tripleta de caracteres compuesta por el carácter “%” seguido por los dos valores hexadecimales (“0123456789ABCDEF”) que conforman el valor hexadecimal del byte (Berners, 1994).

Entre los URL mas usados en la Web se encuentran los empleados por el protocolo HTTP para transmitir páginas Web, la sintaxis de un URL HTTP es:

http://<host>:<puerto>/<ruta>?<cadena de consulta>

Donde podemos observar: el esquema **HTTP**, el **host**; que generalmente es el nombre de dominio del servidor, el puerto **TCP**<sup>30</sup> (80 por omisión), la ruta; utilizada como un seleccionador **HTTP** y en muchos casos su destino es un software generador encargado de la construcción de páginas dinámicas, y la cadena de consulta o “query string”. El “query string” se compone por un conjunto de pares variable-valor que son muy relevantes cuando las páginas son automáticamente generadas, ya que contienen los parámetros específicos de búsqueda que son enviados al software generador; que se encarga de utilizarlos de manera adecuada para la construcción de la página resultante.

En el caso en que se omite la ruta y la cadena de consulta, se utiliza una estructura jerárquica separada por el caracter “/” para designar la ruta a un archivo contenido dentro del servidor Web, y la sintaxis es la siguiente:

`http://<host>:<puerto>/<ruta al archivo>/<archivo>`

## **3.1 Estudio del sistema de consultas**

Una consulta se define en el campo de la informática como una solicitud de información a una base de datos, la misma proporciona una manera de hacer a un lado la información que no se desea ver, para observar con mas claridad la que se necesita (Interactive Programmers Community, 2000). Se caracteriza porque el resultado de la misma, es cero, uno o más registros que contienen la(s) palabra(s) clave consultada(s).

Esta sección estudia las principales características de los sistemas de consultas en los sistemas de repositorios institucionales Alejandría y DSpace.

### **3.1.1 Sistema de consultas de Alejandría**

Como se especifica en el manual de administrador del sistema de repositorio institucional Alejandría (Alejandría, 2005), su sistema de consultas esta formado por un primer conjunto de variables que describen las principales características de un documento, y un segundo conjunto que configuran la salida de los resultados en el navegador. En la figura 7 se muestran las variables mas usadas por el sistema Alejandría para consultar documentos.

---

<sup>30</sup> *Transmission Control Protocol* <http://www.rfc-es.org/rfc/rfc0793-es.txt>

VARIABLE	ESPECIFICACIÓN
Autor	Nombre y apellido del autor de un documento
Título	Título de un documento
Descriptor	Palabra clave que describe a un documento
Forma	Tipo de formulario utilizado para una búsqueda
Fecha	Fecha de publicación de un documento
Grupo	Grupo de usuarios del repositorio
Term_termino_*	Término de una búsqueda compuesta que puede ser autor, título, descriptor y otros
Nombrebd	Nombre de la base de datos que a consultar

Figura 7. Tabla de las variables de búsqueda más usadas por el sistema Alejandría.

Las consultas en Alejandría pueden ser enviadas al servidor por los métodos GET y POST: En el primer caso, las consultas están dispuestas a manera de enlaces electrónicos que se construyen en las páginas estáticas manualmente por el Webmaster, mientras que en las páginas dinámicas; se construyen automáticamente por la aplicación CGI encargada de realizar esta operación. De cualquier modo, se envía al servidor un URL que se compone de tres secciones: la dirección electrónica del servidor Web, el camino a la aplicación CGI que se encarga de manejar la parte dinámica del servidor, y el “query string” o cadena de consulta. Este último, mostrado explícitamente en la barra de direcciones del navegador Web, a su vez, el “query string” está formado por un conjunto de pares variable=valor; que contienen las variables a consultar en el sistema y su correspondiente valor o palabra clave que toma cada una de estas, separados entre si por el caracter “&”, que funge como operador lógico (AND). La sintaxis del URL en este caso es la siguiente:

`http://<direccion del servidor>/<ruta al cgi>?<variable_1=valor_1>&<variable_2=valor_2>...`

En caso de que el método de envío sea POST, las consultas se construyen a partir de una caja de texto como la mostrada en la figura 8, dispuesta en una página del servidor en la que se ingresa manualmente el/los valor(es) de la(s) variable(s) a consultar. Al momento de activar el envío, se construye automáticamente un URL compuesto por dos secciones: La dirección electrónica del servidor Web y el camino a la aplicación CGI que se encarga de manejar la parte dinámica del servidor.

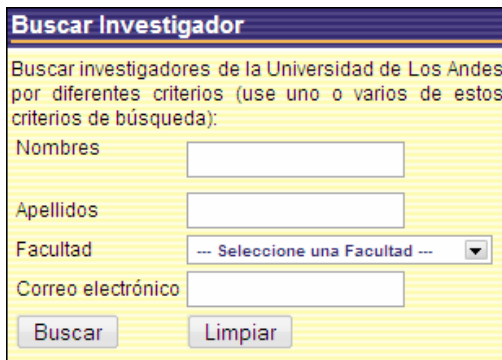


Figura 8. Caja de búsqueda del Servidor SABER-ULA. Tomado de <http://www.saber.ula.ve/unidades/investigadores.html>

Evidentemente, en este caso la petición no contiene “query string”, La sintaxis del URL enviado al servidor Web es la siguiente:

`http://<direccion del servidor>/<ruta al cgi>`

Las peticiones son enviadas al servidor Web codificadas mediante el conjunto de caracteres ISO-8859-1; el servidor las recibe y procesa adecuadamente, dando como resultado una página Web generada dinámicamente que contiene uno o más enlaces a los registros correspondientes a la consulta realizada, la figura 9 es un ejemplo resultante de una consulta al servidor SABER-ULA.



Figura 9. Página resultante de una consulta a SABER-ULA. Tomado de [http://www.saber.ula.ve/cgi-win/be\\_alex.exe?Autor=Acosta+Kanquis,+Nebis](http://www.saber.ula.ve/cgi-win/be_alex.exe?Autor=Acosta+Kanquis,+Nebis)



### 3.1.2 Sistema de consultas de DSpace

El sistema de consultas de la plataforma DSpace se conforma por un conjunto de cajas de búsqueda dispuestas en cada una de sus páginas principales, las cuales acceden a un conjunto definido de metadatos Dublin Core, que almacenan las características principales de un documento, la figura 10 muestra una de esas cajas de búsqueda.

Figura 10. Caja de búsqueda de DSpace. Tomado de <http://dspace.mit.edu/advanced-search>

La figura 11 muestra las principales variables de búsqueda utilizadas por DSpace, así como la especificación de cada una de esas variables.

VARIABLE	ESPECIFICACIÓN
Author	Nombre y apellido del autor de un documento
Title	Título de un documento
Keyword	Palabra clave que identifica a un documento
Subject	Materia a la que pertenece un documento
Abstract	Resumen de un documento
Series	Número de serie en la publicación de un documento
Sponsor	Patrocinante de un documento
Identifier	Números de identificación o códigos asociados a un documento
Language (ISO)	Lenguaje en el que esta escrito un documento

Figura 11. Tabla de las variables de búsqueda del sistema DSpace.

Todas las consultas en DSpace son enviadas al servidor mediante el método GET. Las cajas de búsqueda construyen un URL cuya sintaxis es:

`http://<direccion del servidor>/<ruta al servlet>?<query string>`

Donde se identifican claramente tres secciones: la dirección del servidor, la ruta al cgi y el “query string”, éste último se compone por un único par variable-valor que a su vez posee la siguiente sintaxis:

query=((<variable\_1:valor\_1><operador>(<variable\_2:valor\_2><operador>...)

En él, seguido de la palabra “query” y el caracter “=”, se encierra entre paréntesis la consulta; la misma se forma por un sub-conjunto de pares (variable:valor) encerrados entre paréntesis y separados entre si por uno de los operadores lógicos predefinidos por el sistema (AND, OR y NOT).

DSpace separa sus contenidos en niveles llamados comunidades y colecciones, las cuales se asocian a un identificador único llamado handle, mencionado en la sección 2.2.2 del marco teórico y que se estudiará más a fondo en la sección 3.2. Debido a esto cada nivel contiene un portal con una caja de búsqueda configurada de manera que realice sus consultas al servidor buscando información solo dentro de los registros pertenecientes a la comunidad o colección deseada, estas cajas de búsqueda construyen un URL de la siguiente manera:

http://<direccion del servidor>/<identificador>/<ruta al servlet>?<query string>

El identificador, que es la sección nueva en el URL de la consulta, almacena el handle perteneciente a la comunidad o colección base sobre la cual se realiza la consulta que se encuentra en el “query string”.

Finalmente, utilizando el conjunto de caracteres UTF-8, se codifica el URL y se envía al servidor Web para que la solicitud sea procesada. La respuesta que se arroja como resultado es una página Web compuesta por los enlaces a los registros que concuerden con la búsqueda realizada, tal como lo muestra la figura 12.

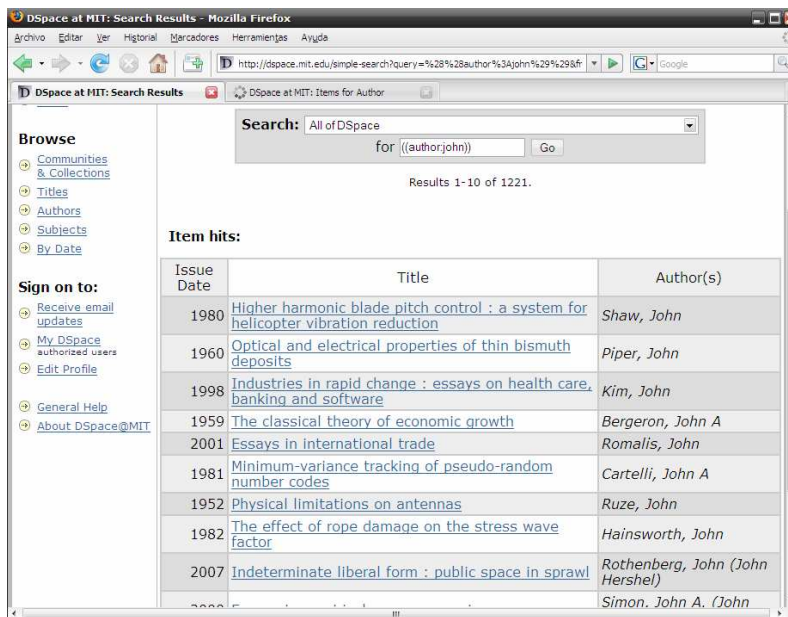


Figura 12. Página resultante de una consulta a DSpace. Tomado de [http://dspace.mit.edu/simple-search?query=\(\(author:john\)\)](http://dspace.mit.edu/simple-search?query=((author:john)))

## 3.2 Estudio del sistema de direccionamiento de documentos

En los sistemas de repositorios institucionales, los documentos se almacenan como un conjunto de metadatos en registros únicos de una misma base de datos. Cada uno de estos registros se asocia a un identificador único en el sistema, el cual es también almacenado como un metadato. El identificador puede ser usado para acceder directamente a un documento en el sistema, fungiendo como direccionador de documentos del repositorio. Esta sección presenta la revisión de los sistemas de direccionamiento de documentos en cada uno de los sistemas de repositorio institucional considerados en el proyecto.

### 3.2.1 Sistema de direccionamiento de documentos de Alejandría

En Alejandría, cada documento posee un identificador único denominado “acceso”. Este sistema de identificación de documentos fue desarrollado en un comienzo para el modelo de bibliotecas digitales de Alejandría, basado en el sistema de información bibliográfica de la CEPAL<sup>31</sup> (Vera, 2003), y posteriormente fue adaptado al sistema de repositorio institucional en estudio. El identificador se compone por cuatro secciones:

<tipo de registro><id biblioteca><primer nivel>/<segundo nivel>

<sup>31</sup> Comisión Económica para América Latina y el Caribe <http://www.cepal.cl/>

- <tipo de registro>: Identifica el tipo de registro de información que se almacena. Alejandría puede almacenar dos tipos de registro de información: documentos (T) y usuarios (U).
- <id biblioteca>: Identificador único de cuatro dígitos asignado a cada biblioteca de la red Alejandría por la empresa HACER-Sistemas.
- <primer nivel>: Identifica el registro de información en un primer nivel de descripción. Es un número único de ocho dígitos asignado secuencialmente por cada biblioteca a los documentos ingresados en ésta.
- <segundo nivel>: Identifica el registro de información en un segundo nivel. Es un número único dentro del primer nivel de identificación de un registro, puede tener uno o dos dígitos.

La siguiente figura, muestra un ejemplo de la identificación de documentos del repositorio institucional SABER-ULA, que funciona bajo el sistema Alejandría:

ACCESO	DESCRIPCIÓN
T016300001134/0	Identificador del registro de información a un primer nivel perteneciente al libro (título, autor, introducción, tabla de contenido, descriptores, enlaces a los capítulos)
T016300001134/1	Identificador del registro de información a un segundo nivel de descripción perteneciente al capítulo 1 (título del capítulo, tabla de contenido del capítulo, enlace al documento)
T016300001134/2	Identificador del registro de información a un segundo nivel de descripción perteneciente al capítulo 2 (título del capítulo, tabla de contenido del capítulo, enlace al documento)
...	...
T016300001134/8	Identificador del registro de información a un segundo nivel de descripción perteneciente al capítulo 8 (título del capítulo, tabla de contenido del capítulo, enlace al documento)

Figura 13. Tabla de los identificadores de registro del libro “Sistemas de conexión a tierra en 60 Hz” en el servidor SABER-ULA.<sup>32</sup>

<sup>32</sup>Tabla construida a partir de un análisis de identificadores de la página Web [http://www.saber.ula.ve/cgi-win/be\\_alex.exe?Acceso=T016300001134/0&Nombred=SSABER](http://www.saber.ula.ve/cgi-win/be_alex.exe?Acceso=T016300001134/0&Nombred=SSABER)

Cualquier registro dentro del repositorio institucional puede ser accedido mediante una consulta; caracterizada por la aparición de una variable cuyo valor es el identificador único dentro del sistema, la figura 14 muestra las variables mediante las cuales se puede acceder directamente a un documento.

VARIABLE	ESPECIFICACIÓN
Acceso	Accede a registros de información de documentos (T).
Ejemplar	
Documento	
Item	
Codigo	Accede a registros de información de usuarios (U).

Figura 14. Tabla de las variables de acceso directo a documentos de Alejandría

La sintaxis del URL utilizado para acceder a un registro es similar a la utilizada para realizar cualquier consulta, solo que al inicio del “query string” aparece una de las variables definidas en la figura 14; a la cual se le asigna como valor un “acceso”, seguida generalmente por la variable que identifica la base de datos a consultar, como se muestra a continuación:

`http://<direccion del servidor>/<ruta al cgi>?<variable=”acceso”>&...`

Este URL se codifica en el conjunto de caracteres ISO-8859-1 y se envía al servidor, dando como resultado una página como la mostrada en la figura 15, con la información del registro solicitado. Además, el resultado se presenta en diferentes formatos dependiendo del a variable utilizada en la consulta, y un enlace al documento.

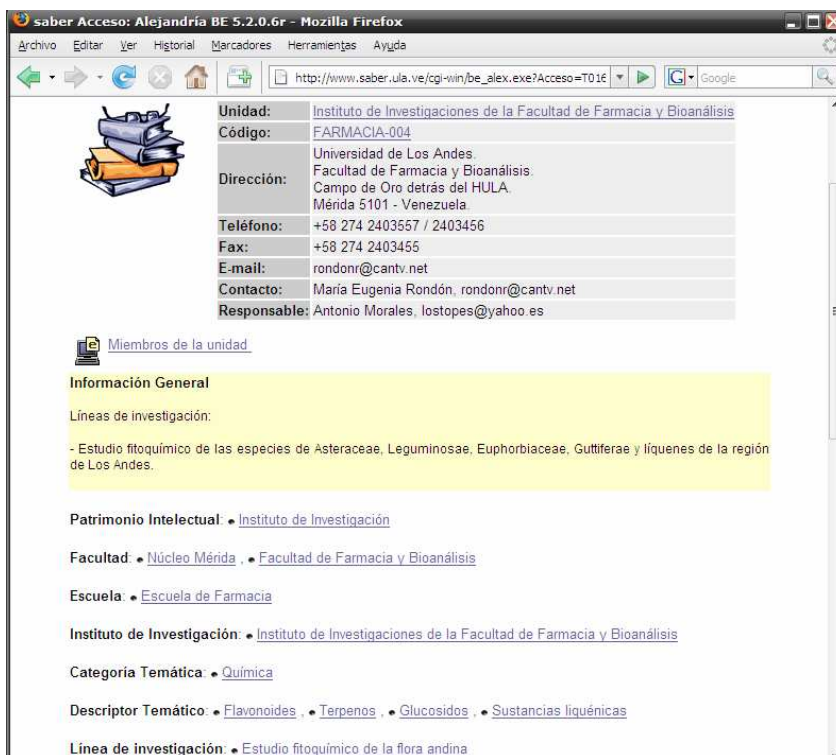


Figura 15. Página resultante de la solicitud de un registro a SABER-ULA. Tomado de [http://www.saber.ula.ve/cgi-win/be\\_alex.exe?Acceso=T016300001447/0](http://www.saber.ula.ve/cgi-win/be_alex.exe?Acceso=T016300001447/0)

### 3.2.2 Sistema de direccionamiento de documentos de Dspace

DSpace identifica sus documentos por medio del Handle System, sistema que como se mencionó en el capítulo 2, forma parte del sistema DOI (Digital Object Identifier) y fue creado para identificar, dentro de los objetos digitales, específicamente las entidades de propiedad intelectual. Cada registro dentro del sistema, almacena como un metadato a su propio identificador denominado “handle”, el cual es único dentro del repositorio, y puede ser único en el mundo si el repositorio se registra ante el sistema de registro del servicio handle<sup>33</sup>

Según el manual técnico del Handle System (Handle.net, 2006), cada identificador se compone de dos partes: la autoridad de nombramiento de handle, conocida como el prefijo, y el nombre único local, también conocido como el sufijo, ambos separados mediante el carácter “/”. El “handle” puede estar compuesto por cualquier carácter imprimible del conjunto de caracteres universal (UCS-2), pero para permitir la compatibilidad con la mayoría de los sistemas existentes, y evitar la ambigüedad entre los diferentes conjuntos de caracteres, el

<sup>33</sup> Resolution Service Providers [http://handle.net/service\\_agreement.html](http://handle.net/service_agreement.html)

protocolo handle codifica el identificador bajo el conjunto de caracteres UTF-8. La sintaxis del identificador se muestra a continuación:

<autoridad de nombramiento de handle>/<nombre único local>

Los registros pertenecientes al repositorio son accedidos mediante un URL compuesto por la dirección electrónica del servidor y el identificador único descrito en el párrafo anterior, separados entre si por la palabra “handle”:

http://<direccion del servidor>/handle/<autoridad de nombramiento de handle>/<nombre único local>

La solicitud devuelve una página Web como la mostrada en la figura 16, conformada por una lista con la información almacenada en los metadatos pertenecientes al registro, así como un enlace al documento referido.

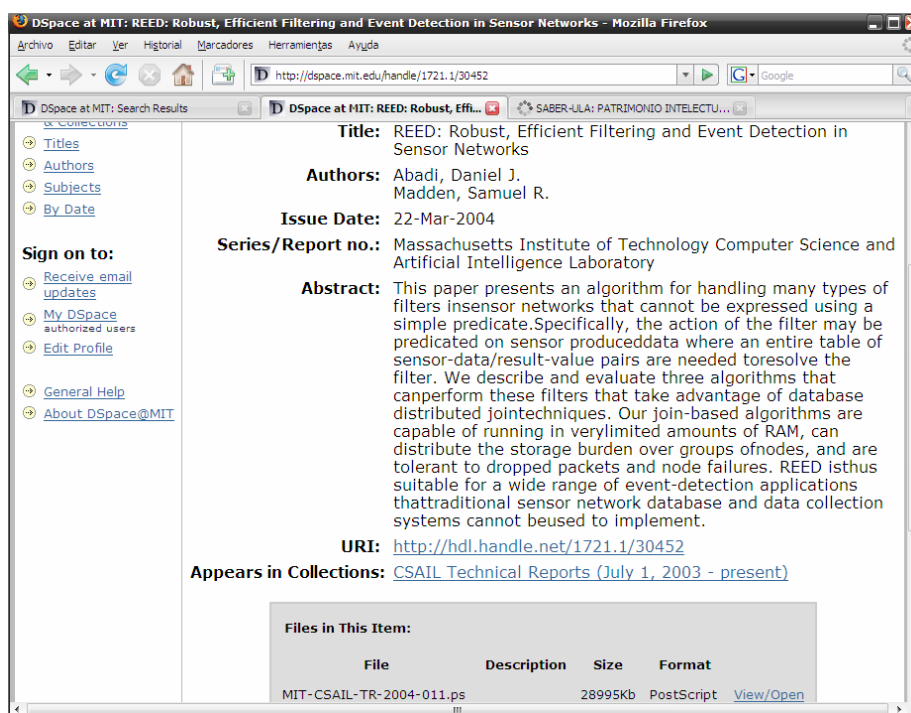


Figura 16. Página resultante de la solicitud de un registro a DSpace. Tomado de <http://dspace.mit.edu/handle/1721.1/30452>

### 3.3 Comparación entre los sistemas de repositorios institucionales

En esta sección se realiza una comparación de las características de los sistemas de consultas y direccionamiento de documentos descritas en el análisis anterior para ambos repositorios institucionales. Se pretenden conocer las semejanzas y diferencias entre ellos, de manera que se pueda formar la base para el diseño del sistema capaz de asociar y convertir cada uno de los URL solicitados al repositorio origen (Alejandría) para redireccionarlos hacia el repositorio destino (DSpace).

#### 3.3.1 Comparación entre los sistemas de consultas

Luego de analizar las características de los sistemas de consultas en los sistemas Alejandría y DSpace, se realizó la siguiente comparación:

1. **Aplicación Generadora:** Ambos sistemas utilizan una aplicación generadora para el manejo de las consultas al servidor y la generación de las páginas dinámicas resultantes. En tanto Alejandría utiliza a la aplicación “cgi-win/be\_alex.exe”, DSpace utiliza el servlet “simple-search” como motor dinámico del servidor.
2. **Método de envío GET:** Tanto Alejandría como DSpace tienen la capacidad de enviar las solicitudes al servidor por medio del método GET; construyendo un URL en el que las variables de consulta son dispuestas en el “query string” como pares variable-valor, con el formato mostrado en la siguiente figura.

ALEJANDRÍA	DSPACE
variable=valor	variable:valor

Figura 17. Tabla comparativa de los pares (variable-valor) de consulta entre los sistemas de repositorios institucionales.

3. **Método de envío POST:** El sistema Alejandría permite el envío de formularios al servidor mediante el método POST, mientras que DSpace no.
4. **Morfología del “query string”:** Alejandría utiliza el estándar HTTP en la construcción del query string, utilizando cualquier cantidad de pares variable-valor. DSpace utiliza solo un par variable-valor (query=<consulta>)), la consulta posee una sintaxis propia del sistema en la que se almacenan entre paréntesis los pares variable-valor.
5. **Operadores lógicos:** Alejandría restringe el uso de los operadores lógicos a la unión de los pares variable-valor mediante el carácter “&” (operador lógico de inclusión), mientras que en DSpace los pares pueden ser asociados entre si por los operadores “AND”,



“OR” y “NOT”. Esta característica permite asociar el carácter de separación de pares de Alejandría “&” con el operador lógico de DSpace “AND”.

6. **Codificación de las solicitudes Web:** Los sistemas no codifican sus solicitudes Web (GET o POST) bajo un mismo conjunto de caracteres: Alejandría los codifica en ISO-8859-1 antes de ser enviados al servidor, en cambio DSpace los codifica en UTF-8. Esto se presenta como una de las diferencias fundamentales en el tratamiento de las solicitudes de consulta por parte de ambos sistemas, dejando como tarea, buscar el mecanismo mediante el cual se puede llevar una solicitud de una codificación a otra, sin modificar los valores de búsqueda.
7. **Variables de consulta:** Se puede realizar una asociación entre algunas de las variables de consulta de Alejandría y DSpace, la cual es mostrada en la figura 18.

ALEJANDRÍA	DSPACE
Autor	Author
Título	Title
Descriptor	Keyword, Subject

Figura 18. Tabla asociativa entre las variables de consulta de Alejandría y DSpace.

Las demás variables del sistema de consultas de Alejandría no poseen una correspondiente variable en DSpace con la que pueda ser asociada, por lo que el conversor se debe limitar a utilizar solo las mostradas en la tabla anterior.

8. **Resultado de la petición:** En ambos sistemas la petición devuelve una página Web compuesta por una lista de los enlaces a los registros de información que concuerdan con la búsqueda realizada.

CARACTERÍSTICA	ALEJANDRÍA	DSPACE
Aplicación Generadora	CGI	Servlet
Método de envío GET	Si	Si
Método de envío POST	Si	No
Morfología del “query string”	variable=valor&... variable=valor	query=((variable:valor)... OPERADOR(variable:valor))
Operadores lógicos	&	AND, OR, NOT
Codificación de las solicitudes Web	ISO-8859-1	UTF-8
Variables de consulta	Autor, Título, Descriptor	Autor, Title, Keyword, Subject
Resultado de la petición	Lista de enlaces	Lista de enlaces

Figura 19. Tabla comparativa de las características principales de los sistemas de consulta de Alejandría y DSpace.

La tabla de la figura 19 presenta el resumen de la comparación entre los sistemas de consulta de Alejandría y DSpace, en ésta se muestran las similitudes y diferencias de las características principales entre ambos sistemas.

### 3.3.2 Comparación entre los sistemas de direccionamiento de documentos

Del análisis anterior de los sistemas de direccionamiento de documento en los sistemas Alejandría y DSpace, se pueden identificar las siguientes características comparativas:

1. **Identificador de registros:** Tanto en Alejandría como en DSpace, los registros de información de los documentos son asociados a un identificador único en el sistema que tiene una estructura definida diferente en cada uno de ellos; pero mediante el cual se puede acceder directamente al documento por medio de una solicitud en la que el URL contiene de alguna manera a ese identificador. En ambos sistemas el identificador es almacenado como metadato dentro del propio registro de información al que se le asocia.
2. **Acceso al registro:** En el sistema Alejandría, se accede al documento mediante una consulta al servidor en la que la variable de búsqueda toma el valor del identificador del registro de información. En DSpace, cada registro se asocia a un “handle” que se dispone en el URL de consulta como una ruta a un documento.
3. **Morfología del identificador:** La siguiente tabla muestra la composición de los identificadores de ambos sistemas, en la que se mantiene para ambos sistemas la identificación del repositorio dentro de una red y la unicidad del identificador dentro de un mismo repositorio.

GENERAL	ALEJANDRÍA	DSPACE
Identificador del tipo de registro	<tipo de registro>	N/A
Identificador del repositorio dentro de una red	<id biblioteca>	<autoridad de nombramiento de handle>
Identificador único local del documento	<primer nivel>	<nombre único local>
	<segundo nivel>	

Figura 20. Tabla comparativa de la composición de los identificadores de registros de información en los sistemas Alejandría y DSpace.

4. **Codificación de las solicitudes Web:** Al igual que en el caso de las consultas, Alejandría codifica las solicitudes bajo el conjunto de caracteres ISO-8859-1, DSpace los codifica en UTF-8, presentándose la misma dificultad para la transformación de dichas peticiones.
5. **Resultado de la petición:** En ambos sistemas la solicitud enviada al servidor da como resultado una página Web que lista la información del registro de información solicitado y un enlace al documento referido.

La tabla presentada a continuación (figura 21) muestra el resumen de las principales características comparativas de los sistemas de direccionamiento de documentos de los RRII en estudio.

CARACTERÍSTICA	ALEJANDRÍA	DSPACE
Identificador de registros	Si	Si
Acceso al registro	Vía “query string”	Vía “ruta”
Morfología del identificador	<tipo de registro><id biblioteca> <primer nivel>/<segundo nivel>	<autoridad de nombramiento de handle>/<nombre único local>
Codificación de las solicitudes Web	ISO-8859-1	UTF-8
Resultado de la petición	Registro único	Registro único

Figura 21. Tabla comparativa de las características principales de los sistemas de direccionamiento de documentos de Alejandría y DSpace.

Del análisis realizado a los sistemas de consulta y direccionamiento de documentos a los sistemas de RRII Alejandría y DSpace, así como de las comparaciones presentadas en las secciones anteriores, se puede tomar una idea general para el desarrollo del sistema. También se presentaron en este capítulo una serie de características y variables que servirán como insumo para el diseño y construcción del conversor.

## Capítulo 4 Diseño

Luego del análisis de los sistemas de repositorios institucionales (RRII) Alejandría y DSpace presentado en el capítulo 3, el presente capítulo muestra la fase de diseño del sistema conversor de mecanismos de consultas y accesos entre los sistemas de RRII ya mencionados. Se presenta una clasificación de las peticiones al servidor realizada de acuerdo a sus características principales, con el propósito de separar en casos el proceso de redireccionamiento. Se presenta un diseño general del funcionamiento del conversor y un diseño detallado de sus componentes: las aplicaciones de redireccionamiento y la configuración y generación de las mismas.

### 4.1 Clasificación de las solicitudes Web

De acuerdo al análisis realizado, se encontraron diversos tipos de solicitudes, cuyas características morfológicas y funcionales hacen que puedan ser clasificadas para optimizar el funcionamiento del conversor.

Se resolvió clasificar las peticiones provenientes del sistema de repositorio origen Alejandría en dos niveles, el primero de éstos es el método de envío de las peticiones al servidor Web: GET y POST. Cada uno de estos niveles es descrito detalladamente en las secciones subsiguientes, en ellas se presenta también el segundo nivel de esta clasificación (figura 22).

PRIMER NIVEL	SEGUNDO NIVEL
GET	C (consulta) FORM (formulario) DOC (documentos) IR (identificador de registro)
POST	POST

Figura 22. Clasificación de las solicitudes Web.

#### 4.1.1 Método GET

El método de envío de peticiones GET descrito en el capítulo 2, se caracteriza porque se muestra en el cliente en la forma de un URL. Entre los URL que realizan las solicitudes al servidor Web del repositorio origen por medio del método GET, se realizó una sub-división de esta clasificación de acuerdo a dos de sus características principales: la morfología del “query string”, y el resultado de la petición. A continuación se presenta esta sub-clasificación:

- **C (consulta):** Este tipo de URL se caracteriza porque contiene un “query string” compuesto por variables que contienen palabras clave para la búsqueda de un documento. Al realizar la petición al servidor Web, el resultado es el conjunto de registros de información contenidos en la base de datos del repositorio; que concuerdan con las palabras clave contenidas en las variables de la búsqueda, por lo que éste resultado puede estar conformado por cero, uno o más registros de información. A continuación se muestra la sintaxis de algunos URL identificados para el caso de estudio Alejandría que entran en esta clasificación:

`http://www.saber.ula.ve/cgi-win/be_alex.exe?Autor=<autor>`

`http://www.saber.ula.ve/cgi-win/be_alex.exe?Titulo=<titulo>`

- **FORM (formulario):** En esta sub-división el “query string” del URL enviado al servidor contiene como primera variable la que identifica al tipo de formulario utilizado para realizar la solicitud (“forma”), seguida del conjunto de variables asociadas a las palabras clave de la búsqueda. Esta petición resulta en un conjunto de cero, uno o más registros de información que concuerdan con las variables de búsqueda presentes en el URL. Como ejemplo se muestra la sintaxis de los siguientes URL solicitados al servidor Web SABER-ULA:

`http://www.saber.ula.ve/cgi-win/be_alex.exe?Forma=FUSU&Fecha=<fecha>`

`http://www.saber.ula.ve/cgi-win/be_alex.exe?forma=FGENERAL&term_termino_1=  
<descriptor_1>& term_termino_2=<descriptor_2>&...`

- **DOC (documento):** Se caracteriza porque el URL no contiene “query string”, en su defecto, contiene una ruta directa a un documento perteneciente al servidor. El resultado de la acción es un documento único almacenado en el directorio asociado a los documentos del servidor Web. En el caso del repositorio institucional SABER-ULA, los documentos se encuentran almacenados en el directorio /db, y la sintaxis de los URL son como la del siguiente ejemplo:

`http://www.saber.ula.ve/db/ssaber/Edocs/tesis/derecho/tesis_uzcatastrid.pdf`

- **IR (identificador de registro):** Este tipo de URL se caracteriza porque contiene un “query string” iniciado con la variable que contiene el valor del identificador único de un registro del sistema. El resultado de la acción solicitada al servidor Web es una lista de los metadatos del registro solicitado, mostrada bajo un formato determinado por la variable contenida en el URL. La sintaxis de los URL que entran en esta clasificación es la siguiente:

```
http://www.saber.ula.ve/cgi-win/be_alex.exe?Acceso=<identificador>
http://www.saber.ula.ve/cgi-win/be_alex.exe?Ejemplar=<identificador>
http://www.saber.ula.ve/cgi-win/be_alex.exe?Codigo=<identificador>
```

#### 4.1.2 Método POST

Dentro de esta clasificación no se encontraron características que pudieran diferenciar las peticiones enviadas al servidor por el método POST, es por esto que en el caso de este método de envío existe solo un tipo de solicitudes:

- **POST:** Se caracteriza porque el método de envío es POST, es decir son formularios. El URL de la petición no contiene “query string”, se limita a enviar la ruta a la aplicación encargada de manejar las consultas al repositorio. El resultado de la acción consta de cero, uno o más registros que contienen las palabras clave contenidas en las variables POST asociadas a la búsqueda. El caso de estudio presenta algunas cajas de búsqueda, a continuación se muestra el código HTML de una de ellas:

```
<!-- Forma de Búsqueda para la revista Aldea Mundo -->
<form method="POST" action="/cgi-win/be_alex.exe">
  <input name="forma" type="hidden" value=FGENERAL>
  <input name="nombrebd" type="hidden" value="Ssaber">
  <input name="tipodoc" value="S" type="hidden">
  (Revista Aldea Mundo)
  <input name="term_termino_1" size=20 class="busqueda">
  <input name="term_campo_1" type="hidden" value="Cualquiera">
  <input name="term_tipo_1" type="hidden" value="Palabras">
  <input name="term_termino_2" type="hidden" value="Aldea Mundo">
  <input name="term_campo_2" type="hidden" value="Materia">
  <input name="term_tipo_2" type="hidden" value="Frase">
```

```

<input name="encab" type="hidden" value="0">
<input type="submit" class="busqueda" value="Buscar!" name="submit">
</form>

```

El URL que construye la caja de búsqueda de la revista Aldea Mundo<sup>34</sup> perteneciente al servidor Web SABER-ULA es el siguiente:

[http://www.saber.ula.ve/cgi-win/be\\_alex.exe](http://www.saber.ula.ve/cgi-win/be_alex.exe)

Como ya se explicó, el URL no contiene “query string”, en el ejemplo solo se pueden identificar: El host, y la ruta a la aplicación CGI. Los datos de la búsqueda viajan en el cuerpo de la petición.

## 4.2 Diseño del conversor

Esta sección muestra el diseño del conversor de mecanismos de consultas y accesos entre RRII, modelando las acciones del sistema en el lenguaje UML<sup>35</sup> (versión 2.0), bajo el uso de la herramienta Paccstar UML Diagrammer<sup>36</sup> (Versión 6.02.2014). El sistema conversor se diseñó en base a las características funcionales de un repositorio institucional (RI), con el fin de sustituir la aplicación CGI que funciona como interfaz entre el servidor Web y el sistema Alejandría. Este conversor será el encargado de redireccionar las solicitudes Web enviadas al RI origen, hacia el RI destino (figura 23).

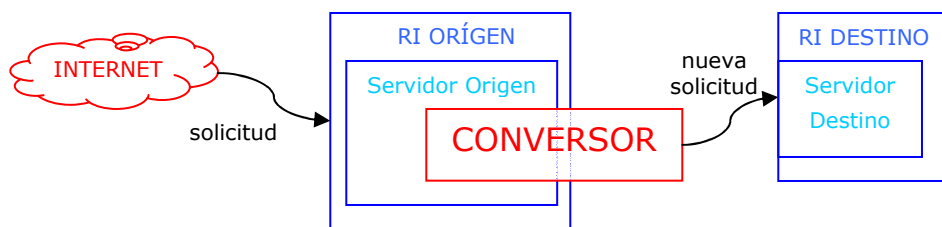


Figura 23. Diagrama funcional del conversor.

El conversor posee dos componentes principales (figura 24): las aplicaciones de redireccionamiento y el módulo de configuración y generación de estas aplicaciones. El diseño

<sup>34</sup> Revista Aldea Mundo <http://www.saber.ula.ve/cefi/aldeamundo>

<sup>35</sup> *Unified Modeling Language* <http://www.uml.org>

<sup>36</sup> *Paccstar Software* bajo licencia de prueba número 5236-6022014-482B336E-000000000-0000-0000 <http://www.paccstar.com/uml>

del conjunto de aplicaciones encargadas de realizar la reescritura y redireccionamiento de las solicitudes solicitados al servidor se explica en la sección 4.2.1, además, en la sección siguiente se detalla el diseño de la configuración del convertor y del generador de esas aplicaciones de redireccionamiento.



Figura 24. Diagrama de componentes del sistema convertor.

En general, un sistema de repositorio institucional esta estructurado de acuerdo al diagrama mostrado en la figura 25:

- El servidor Web: encargado de manejar las transacciones HTTP entre los usuarios de Internet; que acceden a la información por medio de su dirección electrónica, y los datos almacenados en la base de datos del repositorio.
- La aplicación generadora: encargada de gestionar las consultas al servidor y generar las páginas Web dinámicas.
- El manejador de bases de datos: encargado de la interacción entre la aplicación generadora y la base de datos que almacena la información.

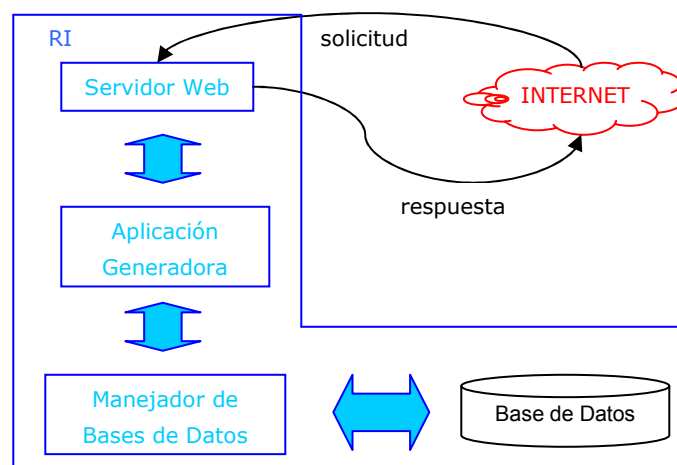


Figura 25. Estructura de un sistema de repositorio institucional.

Tomando como base el funcionamiento de esta estructura, se diseñó un sistema con la finalidad de capturar las solicitudes realizadas al RI origen (Alejandría) por medio del servidor



Web, compararlos con la clasificación descrita en la sección 4.1, y dirigirlos adecuadamente hacia una aplicación encargada de reescribir y redireccionar las nuevas solicitudes adaptadas al sistema destino. Este sistema, cuya estructura se muestra en la figura 26, es el llamado conversor de mecanismos y accesos entre sistemas de repositorios institucionales.

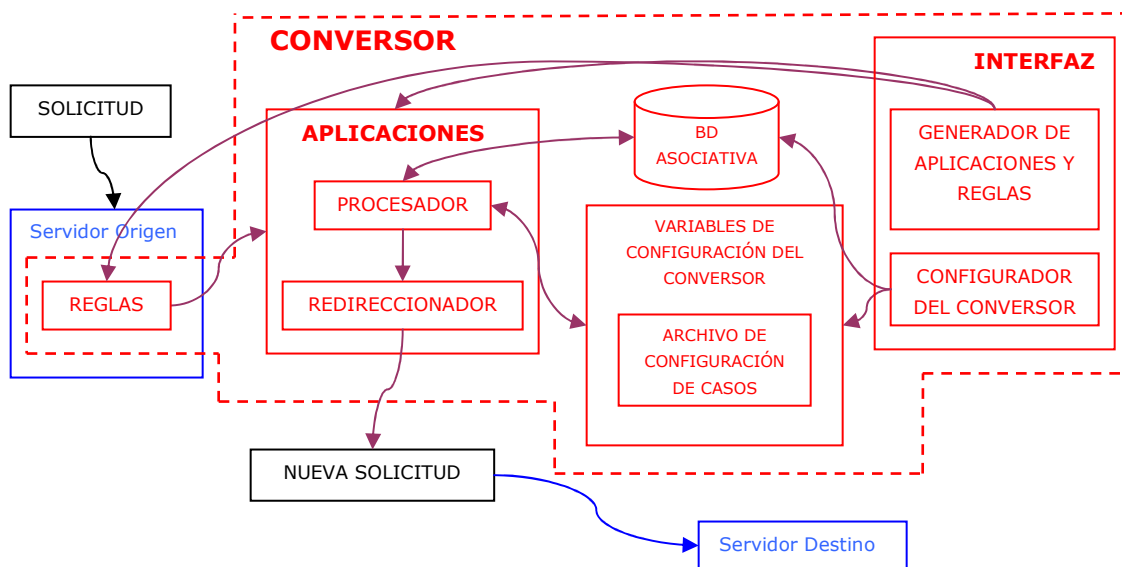


Figura 26. Estructura del conversor.

El conversor está compuesto por un conjunto de reglas encargadas de comparar y dirigir las solicitudes provenientes del servidor origen hacia una de las aplicaciones de reescritura y redireccionamiento. Esta aplicación la recibe y tiene como tarea procesar la solicitud por medio de diversos mecanismos para adaptarla al sistema destino. De acuerdo al análisis realizado en el capítulo 3, es fundamental para esta adaptación el uso de una base de datos (BD) asociativa; que se encargue de enlazar algunas características funcionales de los sistemas. Finalmente la aplicación debe encargarse de redireccionar la nueva solicitud al servidor destino. Adicionalmente, para facilitar el uso del conversor, el sistema debe generar automáticamente a través de una interfaz el conjunto de reglas, aplicaciones de redireccionamiento y la BD asociativa. Tanto el diseño de las aplicaciones, como el de la configuración y generación del conversor se describen en las siguientes sub-secciones.

#### 4.2.1 Diseño de las aplicaciones de redireccionamiento de solicitudes Web

Al presentar el diseño del sistema conversor se mencionó que el conversor está compuesto en parte por un conjunto de aplicaciones encargadas de la reescritura y redireccionamiento de las

solicitudes al servidor Web. Esta sección muestra entonces, el diseño de esas aplicaciones, los casos de uso y diagramas de actividades para cada una de ellas, así como la descripción del proceso que realiza cada aplicación luego que recibe una solicitud, para reescribirla y redireccionarla adecuadamente.

ACTORES	DESCRIPCIÓN	CASOS DE USO
Usuario	Realiza las solicitudes al servidor a través del cliente, quien las envía al servidor para que las clasifique y las dirija hacia la correspondiente aplicación de redireccionamiento.	Redireccionar consultas Redireccionar accesos Redireccionar documentos

Figura 27. Tabla de actores y casos de uso del diseño de las aplicaciones de redireccionamiento.

Como se muestra en la figura 27, desde el punto de vista del usuario, se presentan tres tipos de enlaces, por lo tanto tres casos de uso en el redireccionamiento de solicitudes que se describen a continuación: las consultas, los accesos y los documentos.

### Redireccionar consultas

El caso de uso mostrado en la figura 28 ilustra las acciones que deben realizar el servidor y la aplicación encargada de redireccionar las solicitudes clasificadas como una consulta al repositorio

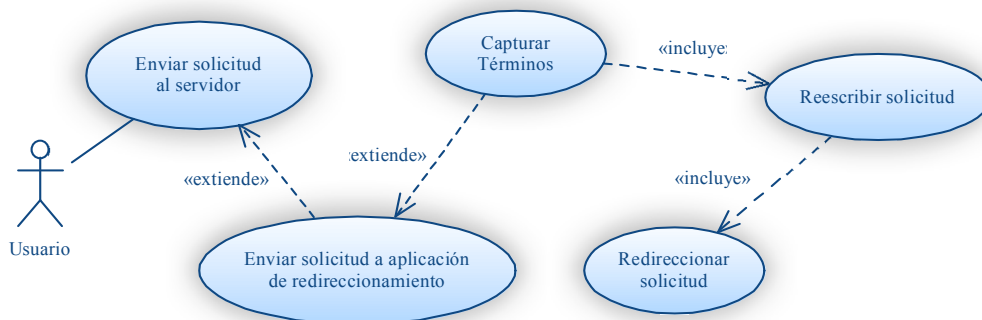


Figura 28. Caso de uso “redireccionar consulta”.

Luego que el usuario envía la solicitud al servidor Web, el cual la clasifica como una consulta y la envía hacia la aplicación correspondiente, esta captura los términos de la consulta y los reescribe en una nueva solicitud que es adaptado al sistema de consultas del RI destino

(DSpace) de acuerdo a sus características morfológicas y funcionales. Finalmente se redirecciona la petición enviando un nuevo URL al servidor correspondiente, la figura 29 presenta un diagrama de las actividades descritas.

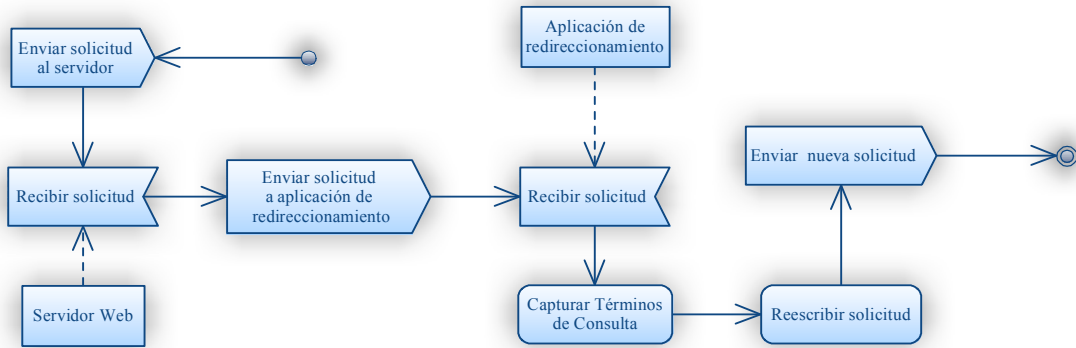


Figura 29. Diagrama de actividades del caso de uso “redireccionar consulta”.

Dentro de este caso de uso podemos encontrar los tipos de solicitudes: C, FORM y POST, de la clasificación descrita en la sección 4.1.

### Redireccionar accesos

De la clasificación de las solicitudes presentada en la sección 4.1, se incluyen en este caso de uso las solicitudes de tipo IR. Como se muestra en el caso de uso de la figura 30, luego que el servidor Web recibe la solicitud, los accesos deben ser capturados de la solicitud origen por la aplicación de redireccionamiento, ser asociados con el identificador correspondiente en el sistema destino, para finalmente reescribir y redireccionar la nueva solicitud.

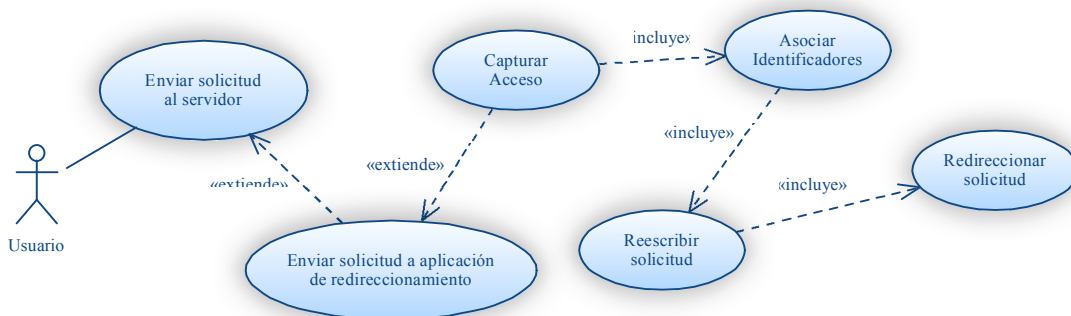


Figura 30. Caso de uso “redireccionar acceso”.

El usuario envía la solicitud, luego el servidor Web envía la solicitud previamente clasificada como acceso hacia una aplicación de redireccionamiento, esta lo recibe y captura en una variable el valor del identificador del registro de información en el sistema origen. Luego este identificador es consultado en una base de datos que asocia los identificadores del sistema origen con los correspondientes en el sistema destino; si la consulta retorna el identificador asociado, se reescribe la solicitud y se redirecciona hacia el nuevo sistema de RI. En caso contrario, se solicita una página por omisión que de alguna manera pueda introducir al usuario en el nuevo sistema, y prestarle las herramientas de búsqueda necesarias para satisfacer su solicitud manualmente.

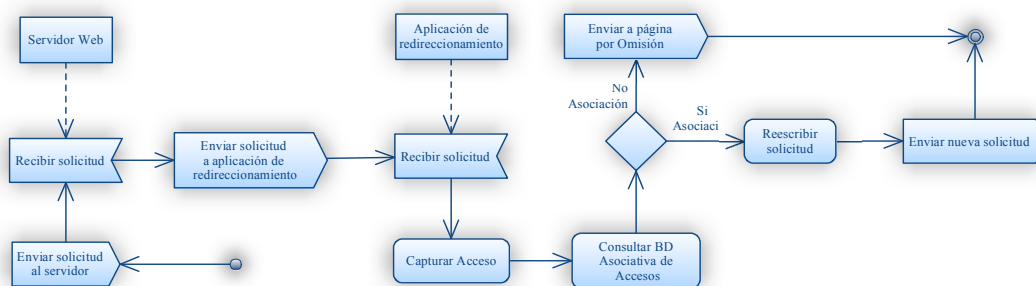


Figura 31. Diagrama de actividades del caso de uso “redireccionar acceso”.

Como se ilustra en la figura 31, el proceso de reescritura de la solicitud Web se fundamenta en una consulta a la base de datos que asocia los identificadores de los sistemas de RRII en estudio. Esta base de datos debe ser creada por el administrador del sistema en el proceso de configuración del sistema conversor, el diseño de este proceso se detalla en la sección 4.2.2.

### Redireccionar documentos

Los documentos a los que acceden los usuarios del repositorio se encuentran almacenados en un directorio del servidor Web, cada solicitud realizada por un usuario consta de un URL con la ruta al documento dentro de ese servidor. En el momento en que el servidor determina que se esta solicitando un documento, el URL solicitado debe ser enviado a una aplicación encargada de capturar esta ruta, asociarla al registro de ese mismo documento en el sistema destino, y redireccionar por medio de un nuevo URL la solicitud. El caso de uso que se muestra en la figura 32, se asocia a las solicitudes de tipo DOC presentadas en la clasificación de solicitudes Web (sección 4.1).

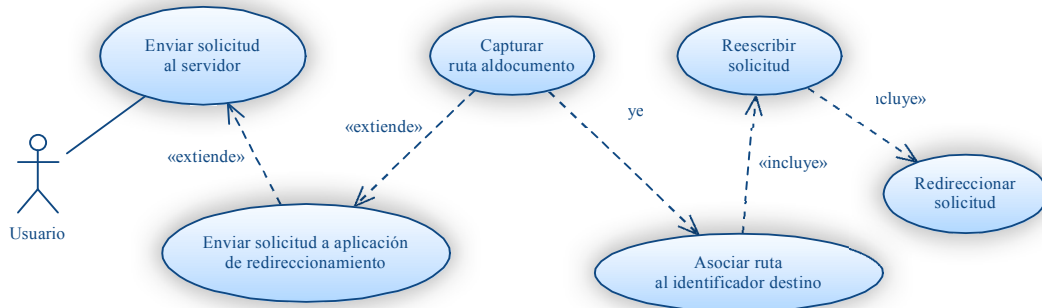


Figura 32. Caso de uso “redireccionar documento”.

La figura 33 ilustra por medio de un diagrama de actividades el caso de uso “redireccionar documento” (figura 32), se puede observar el flujo de las acciones realizadas por el sistema conversor luego que el servidor Web dirige la solicitud hacia la aplicación de redireccionamiento correspondiente. Al igual que en los accesos, se hace necesaria la consulta a una base de datos creada por el administrador del sistema; que asocia las rutas de los documentos en el sistema de RI origen con los identificadores de los registros de información de ese mismo documento en el sistema destino.

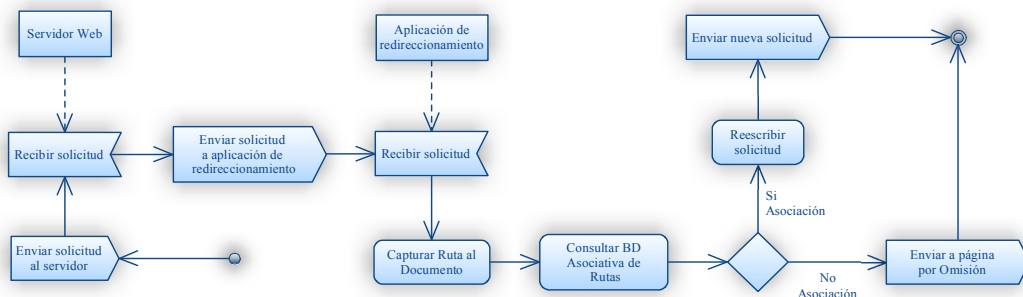


Figura 33. Diagrama de actividades del caso de uso “redireccionar documento”.

La nueva solicitud se reescribe adaptado al sistema destino dependiendo del resultado de esa consulta, en caso de ser positivo, la solicitud se transforma en un acceso al registro de información almacenado en el repositorio destino.

#### 4.2.2 Diseño la de configuración y generación de las aplicaciones de redireccionamiento del sistema

Como se mencionó en la sección 4.2, otra de las partes que componen el sistema conversor es la encargada de configurarlo y generar las aplicaciones y reglas de redireccionamiento. En primer lugar se decidió diseñar como valor agregado, una interfaz gráfica de usuario, con el

propósito de facilitar el uso del conversor. El proceso de diseño se realizó de manera progresiva, en él participaron actores como el administrador del sistema y el asesor del proyecto, aportando ideas y conocimiento acerca del desarrollo y la mejor adaptación al usuario final de este producto, lo que resultó en un diseño original adaptado al caso de estudio. La interfaz posee un diseño sencillo orientado al Web, en el que destacan dos secciones (figura 34): La izquierda, que contendrá las herramientas para accionar la configuración del sistema y generación de aplicaciones y reglas, y la derecha que funcionará como objetivo de visualización de las acciones realizadas.

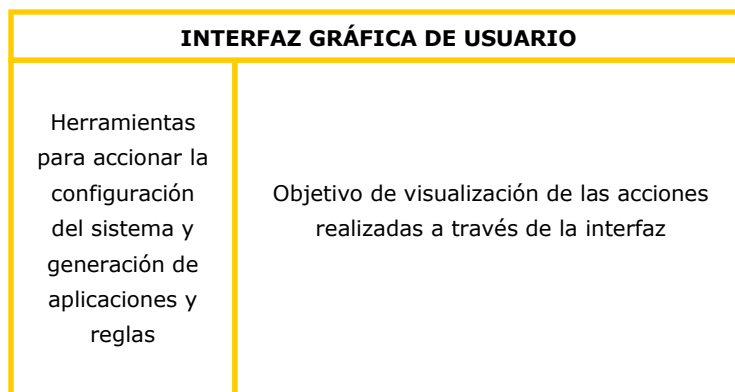


Figura 34. Diseño general de la interfaz gráfica de usuario

La sección izquierda de la interfaz se conformará de los tres siguientes grupos de herramientas:

- Ayuda: para la instalación, configuración y uso del sistema.
- Manejo de la BD asociativa: creación, copiado y eliminación de tablas en la BD.
- Generación de Archivos y Reglas: manejan el módulo de generación de archivos y reglas de redireccionamiento.

A continuación se presenta el diseño de un conjunto de herramientas, entre las cuales tenemos: instalación y configuración del sistema, manejo de la base de datos (BD) asociativa y generación de aplicaciones y reglas de redireccionamiento. Que desde el punto de vista del administrador del sistema, quien funge como actor, define los casos de uso presentados en la tabla de la figura 35.

ACTORES	DESCRIPCIÓN	CASOS DE USO
Administrador	Se encarga de instalar el sistema, configurarlo para su uso y generar las aplicaciones y reglas de redireccionamiento.	Instalar conversor Configurar conversor Manejar BD asociativa Generar aplicaciones y reglas de redireccionamiento

Figura 35. Tabla de actores y casos de uso del diseño de la configuración y generación de las aplicaciones de redireccionamiento.

### Instalar conversor

Como se muestra en la figura 36, el administrador del sistema es el encargado de su instalación, el proceso debe ser rápido y sencillo, y para colocar en funcionamiento el conversor debe ser configurado adecuadamente al ambiente que lo rodea. También se deben generar las aplicaciones y reglas que fungirán como estructura funcional del conversor.

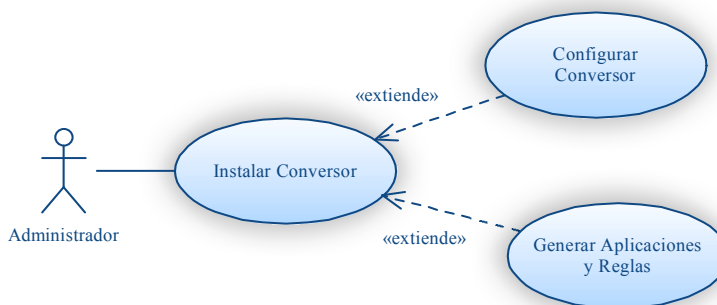


Figura 36. Caso de uso “instalar conversor”.

Como cualquier proceso de instalación de un sistema, este debe comenzar con la creación de directorios y descompresión del paquete, pero además de estas acciones básicas deben existir dos cosas:

- Una herramienta de configuración del sistema: Debido a que el conversor debe adaptarse a las características y variables de dos sistemas de RRII (origen y destino), el proceso de configuración es determinante, es por esto que se diseña una herramienta que minimice el error humano al momento de la configuración y evite la realización de acciones repetitivas dentro de este proceso.
- Una aplicación automatizada de generación de aplicaciones y reglas de redireccionamiento: En vista que existe la posibilidad de encontrar similitudes en el funcionamiento de algunos casos de redireccionamiento, además de que en el proceso

de adaptación y pruebas al sistema se pueden identificar mas casos de los iniciales, y la codificación de las aplicaciones depende de las variables configuradas al instalar el sistema: se debe crear una aplicación, que recibiendo como entrada esas variables y un conjunto de plantillas; sea capaz de generar las aplicaciones y reglas de redireccionamiento del conversor.

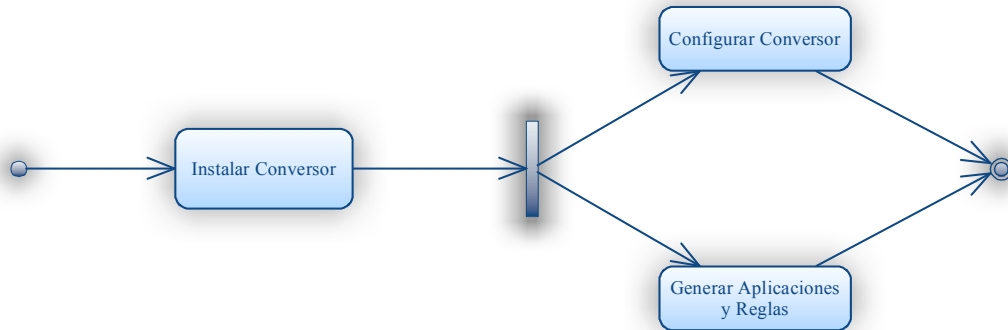


Figura 37. Diagrama de actividades del caso de uso “instalar conversor”.

El diagrama de la figura 37 ilustra las actividades del proceso de instalación del sistema conversor de mecanismos de consultas y accesos entre sistemas de RRII.

### Configurar conversor

Como ya se mencionó en esta sección, se diseñó una herramienta de configuración del sistema. La herramienta consta de tres módulos de configuración que se diseñaron en base al caso de uso de la figura 38: la configuración de las variables de uso del sistema, la configuración de los casos de redireccionamiento identificados por el administrador y el manejo de la BD asociativa.

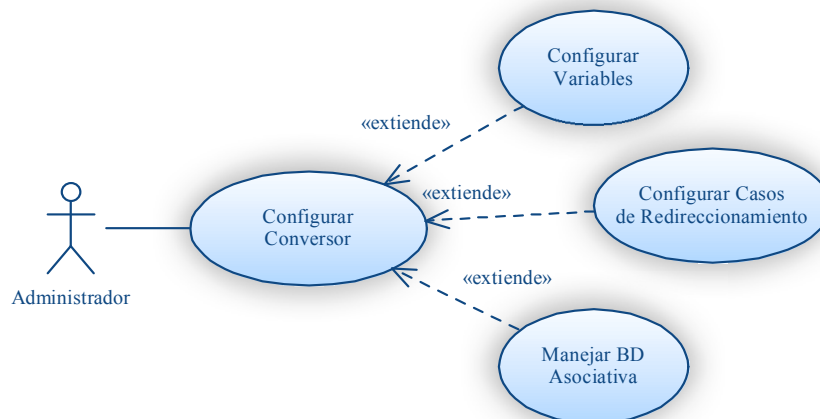


Figura 38. Caso de uso “configurar conversor”.



La configuración de las variables de uso del sistema se realiza mediante la modificación de un archivo de configuración que contiene la lista de las variables necesarias para el funcionamiento de todo el sistema conversor. Igualmente los casos de redireccionamiento se especifican mediante la modificación de un archivo de configuración. La configuración del sistema conversor para su uso no esta completa sin antes configurar la base de datos asociativa, el diseño del módulo de manejo de esta BD se explica mas adelante en detalle. El diagrama de la figura 39 hace referencia a las actividades del caso de uso “configurar conversor” descritas en este párrafo.

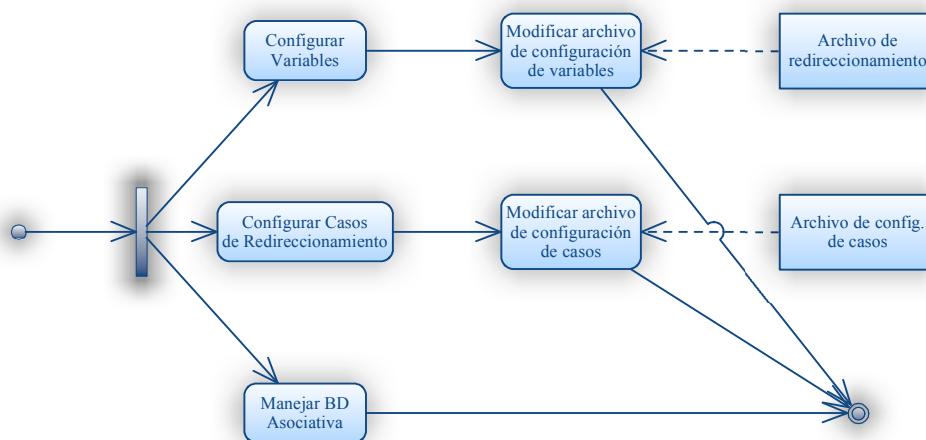


Figura 39. Diagrama de actividades del caso de uso “configurar conversor”.

### Manejar la base de datos asociativa

En diversas secciones se ha mencionado la necesidad de una base de datos asociativa para el funcionamiento del conversor, específicamente para los casos de redireccionamiento en que se deben enlazar ambos sistemas de RI en estudio por medio de alguna característica. La importancia de esta base de datos radica en la asociación entre los sistemas de RRII, la figura 40 muestra las características asociativas identificadas para el sistema origen (Alejandría).

CARACTERÍSTICAS
Identificador de un registro de información
Ruta a un documento
Nombre de un grupo o comunidad

Figura 40. Características asociativas para el sistema de RI origen (Alejandría).

Se tomaron en consideración estas características del sistema origen para asociarlas al sistema destino debido a que son valores (y no variables), los que difieren en cada sistema en estudio, por lo que no puede realizarse una adaptación como en el caso de las variables de consulta (por ejemplo). En este caso, debe realizarse una sustitución del valor original por el equivalente en el sistema destino, de las posibilidades para realizar esta acción se escogió el uso de una base de datos que asocie los valores (origen y destino) y por medio de una consulta devuelva el valor correspondiente.

La figura 41 muestra un ejemplo del caso en que debe asociarse una misma característica que toma valores diferentes en los sistemas en estudio, allí se ilustra como el identificador de un registro de información del sistema origen (acceso, Alejandría) se asocia al identificador correspondiente en el sistema destino (handle, DSpace) para adaptar la nueva solicitud al RI destino.

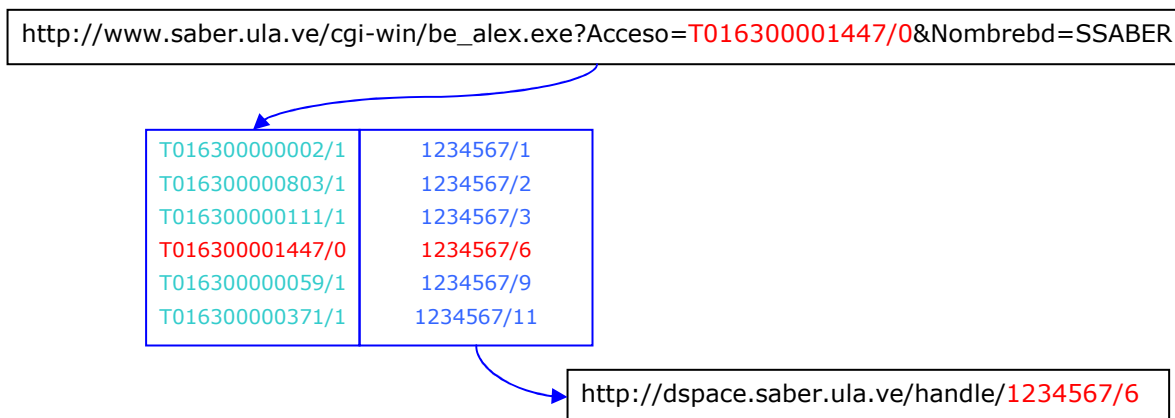


Figura 41. Esquema de la asociación de un acceso (Alejandría) con un handle (DSpace).

Los valores asociados a las características identificadas en la figura 40, dependen en el sistema destino de la lógica funcional de las aplicaciones de redireccionamiento, así como del funcionamiento del conversor, por lo que no siempre se asociarán los valores de las variables equivalentes. Existen casos en los que es más conveniente asociar valores que devuelvan un mejor resultado a la solicitud, de allí resulta la asociación presentada en la siguiente figura.

CARACTERÍSTICA ORIGEN	CARACTERÍSTICA DESTINO
Identificador de registro (acceso)	Identificador de registro (handle)
Ruta al documento	Identificador de registro (handle)
Nombre de grupo o comunidad	Identificador de comunidad (handle)

Figura 42. Tabla asociativa de las características entre los RRIL.

A raíz de los resultados mostrados en la figura anterior, se decidió diseñar una base de datos que contendrá tres tablas:

- Una tabla que asocia el identificador del registro de información de un documento en el sistema de RI origen, con el correspondiente identificador del documento en el sistema de RI destino. A la tabla se le denominó “asociacion\_id”, y posee dos campos: id\_origen e id\_destino, cada uno almacena el correspondiente identificador ya mencionado.
- Una tabla que asocia el nombre de una comunidad o grupo de documentos en el repositorio origen, con el correspondiente identificador en el sistema destino. La tabla “comunidad\_id” esta conformada por: el campo “comunidad\_origen” que almacena el nombre de la comunidad y el campo ”id\_destino” que almacena su correspondiente identificador.
- Una tabla que asocia la ruta de un documento en el repositorio origen, con el identificador de su registro de información. El campo “url\_archivo” almacena la ruta a un documento en la tabla “archivo\_id”, mientras que el campo “id\_origen” almacena el correspondiente identificador del documento en el sistema origen.

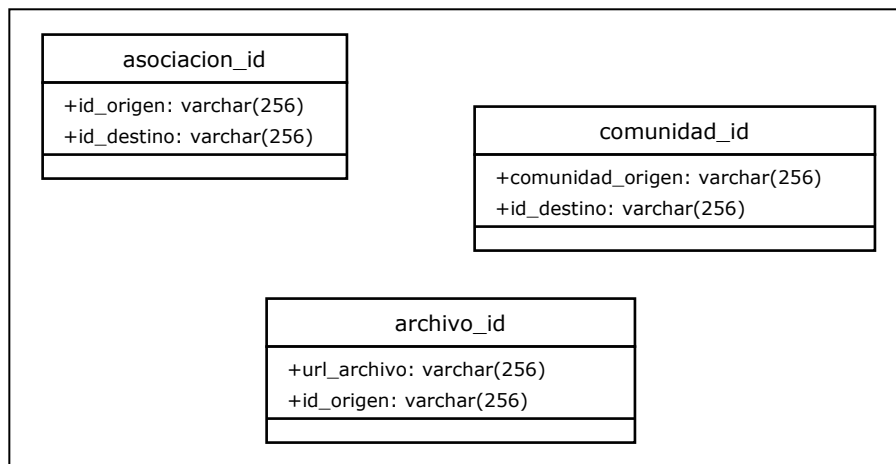


Figura 43. Diseño conceptual de la base de datos asociativa

La figura anterior presenta el diseño conceptual de la base de datos, de la cual se produce como resultado el diseño implementable presentado a continuación en la figura 44.

NOMBRE	ATRIBUTOS
asociación_id	( <u>id origen</u> , id_destino)
comunidad_id	( <u>comunidad origen</u> , id_destino)
archivo_id	( <u>url archivo</u> , id_origen)

Figura 44. Diseño implementable de la base de datos.

Aun cuando el manejo de la base de datos asociativa se presenta como un sub-caso de la configuración del sistema, es de por sí un módulo completo, diseñado en base a las características de los sistemas de consultas y direccionamiento de documentos analizados en el capítulo 3. El caso de uso “manejar BD asociativa” (figura 45) ilustra las acciones que debe realizar este módulo del sistema, el cual se diseñó de modo que se puedan crear, llenar y eliminar un grupo de tablas a las cuales se accederá cuando entre en funcionamiento la aplicación correspondiente del conversor de mecanismos de consultas y accesos.

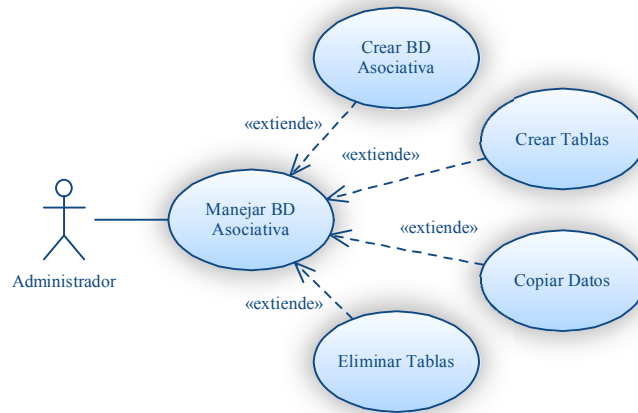


Figura 45. Caso de uso “manejar BD asociativa”.

El módulo de manejo de la BD se diseñó de tal manera que se puedan crear automáticamente las tablas de la BD que van a contener los términos que asocian las características de los sistemas en estudio, exista la posibilidad de copiar los datos en las tablas asociativas, y finalmente permita eliminar las tablas y sus datos (figura 46).



Figura 46. Esquema entrada-salida del módulo de manejo de la BD asociativa.

Como se muestra en el diagrama de la figura 47, algunas de las acciones en el manejo de la BD implican la conexión a la misma para el envío de la sentencia SQL que la realizará.

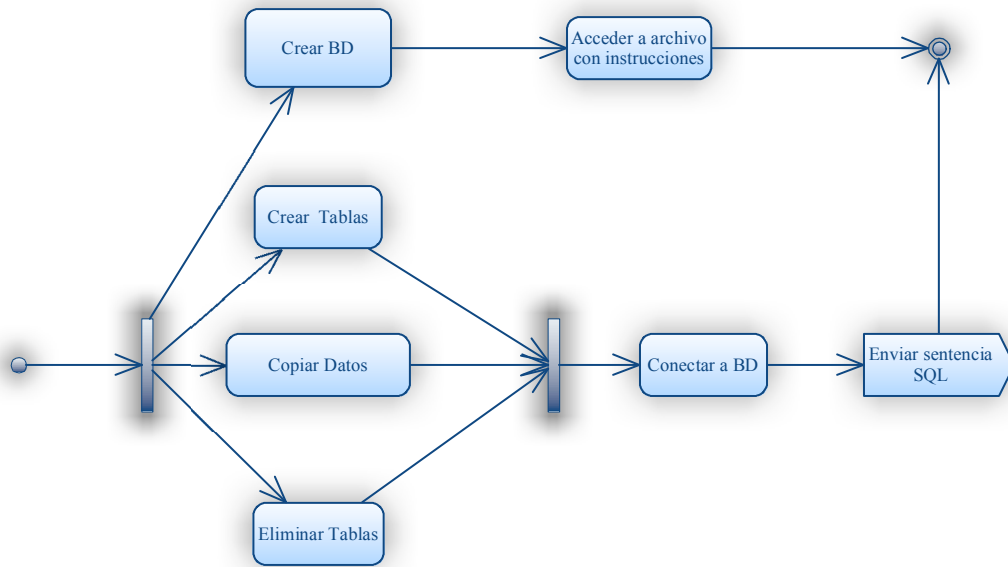


Figura 47. Diagrama de actividades del caso de uso “manejar BD asociativa”.

La especificación de las tablas y los campos de cada una de ellas se definen en la sección 5.1, donde se explica la etapa de implementación del sistema.

### Generar aplicaciones y reglas de redireccionamiento

Como se mencionó al inicio de la sección, se diseñó una aplicación generadora de las aplicaciones y reglas de redireccionamiento que utiliza el convertor. El administrador del sistema es el encargado de activar esta aplicación (figura 48), para esto el sistema convertor debe estar completamente configurado, lo que implica: configurar las variables, configurar los casos y configurar la BD asociativa.

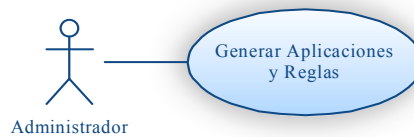


Figura 48. Caso de uso “generar aplicaciones y reglas de redireccionamiento”.

Al activar la aplicación, esta tomará como entrada el archivo de configuración de los casos de redireccionamiento y utilizará algunas de las variables que se encuentran en el archivo de configuración de variables previamente configurado. De acuerdo al diseño, la salida debe estar formada por el conjunto de aplicaciones de redireccionamiento que utilizará el convertor,

y por separado, un archivo de texto con las reglas de redireccionamiento que utilizará el servidor Web para enviar las nuevas solicitudes hacia cada una de las aplicaciones generadas (figura 49).



Figura 49. Esquema entrada-salida del módulo de generación de aplicaciones y reglas de redireccionamiento.

El flujo de las actividades del módulo de generación diseñado se muestra en el diagrama de la figura 50.

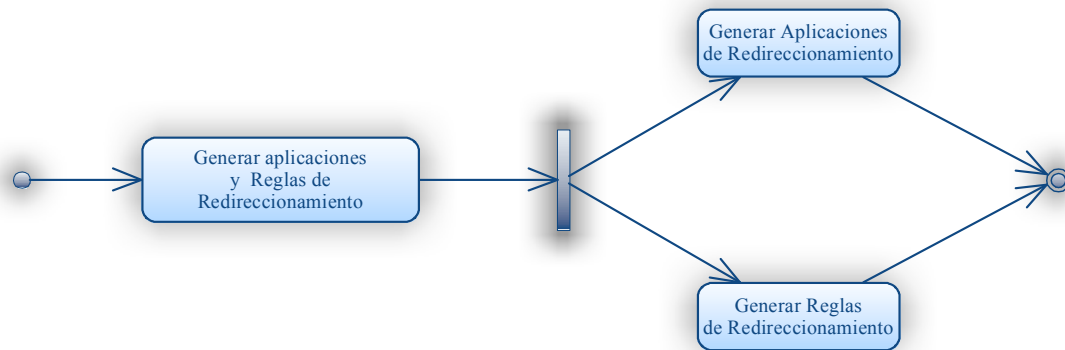


Figura 50. Diagrama de actividades del caso de uso “generar aplicaciones y reglas de redireccionamiento”.

El diseño presentado en este capítulo, es la base para la implementación (presentada en el capítulo 5) del sistema convertor de mecanismos de consultas y accesos entre los sistemas de RRII Alejandría y DSpace. También se verificarán los diferentes casos de uso presentados por medio de algunas pruebas descritas mas adelante. Es importante hacer notar entonces, la gran utilidad del diseño de la aplicación para las fases consecuentes del desarrollo del sistema.

## Capítulo 5 Implementación

La implementación del sistema presentada a continuación se realizó en base al diseño mostrado en el capítulo 4. Este capítulo describe las características de la fase de implementación del conversor, la aplicación del sistema implementado al caso de estudio (Alejandría-DSpace) y finalmente el proceso de refinamiento incremental por versiones y las diferentes pruebas realizadas al conversor.

### 5.1 Implementación del sistema conversor

La sección que se presenta a continuación, describe la implementación del conjunto de módulos, aplicaciones y herramientas que forman parte del conversor. Los componentes y sub-componentes presentados, se describen de manera separada para facilitar la comprensión de la implementación del sistema por parte del lector. Por esta razón se dividió la presente sección describiendo separadamente la implementación de las aplicaciones de redireccionamiento, las herramientas utilizadas para el redireccionamiento, el sistema empleado para implementar la base de datos, la estructura del archivo de configuración de los casos de redireccionamiento, la implementación de la interfaz gráfica de usuario y el contenido del manual de administración del sistema.

#### **Aplicaciones de redireccionamiento**

Las aplicaciones de redireccionamiento se implementaron en el lenguaje de programación PHP (versión 5). Entre las razones por las que se escogió el lenguaje PHP para codificar las aplicaciones de redireccionamiento tenemos que: tiene la capacidad de manejar conexiones a bases de datos, incrustarse e interactuar con archivos codificados en el lenguaje HTML, manejar archivos de texto, manejar las variables de envío y recepción de datos del protocolo HTTP y las variables de ambiente del servidor Web, realizar operaciones con cadenas de caracteres, entre otras.

Se codificó en el lenguaje de programación PHP una plantilla general para cada tipo de solicitud Web definida en la sección 4.1. En el momento en que se configuren los casos de redireccionamiento y las variables de uso del sistema, se procederá a accionar el módulo de generación de aplicaciones y reglas, el cual utilizará estas plantillas como base para la construcción automatizada y particular al caso de las aplicaciones de redireccionamiento. Cada

plantilla será completada con las variables de configuración del sistema y las correspondientes características del caso de redireccionamiento, para así generar las aplicaciones encargadas de reescribir y redireccionar las solicitudes Web. A continuación se muestra el código de una de estas plantillas, ya particularizada con el caso 7, para el caso de redireccionamiento de solicitudes de tipo IR (ejemplo [http://www.saber.ula.ve/cgi-win/be\\_alex.exe?Acceso=T016300001447/0](http://www.saber.ula.ve/cgi-win/be_alex.exe?Acceso=T016300001447/0)):

```
<?php
// Programa llamado desde la regla mod_rewrite
// Caso 7: Acceso directo al documento via ACCESO

include("../conf/conf.txt"); // Archivo de variables de uso

$db= pg_connect($conexion_bd);

if($db){
    $resultado=$_REQUEST["Acceso"];

    $query = "SELECT *
FROM asociacion_id
WHERE id_origen = '$resultado'";

    $result = pg_query($db, $query);

    if(pg_numrows($result) > 0){

        $row = 0;

        $row = pg_fetch_row ($result, 0);

        $url = $base_dir.$row[1];

    }else{
        $url = $url_destino;
    }
}
pg_close($db)
?>
<?php
header("Location: $url");

?>
```

El código mostrado se divide en dos secciones. La primera es el código PHP, que se encarga de tomar el valor de la variable “acceso”, buscar su identificador correspondiente en el sistema destino por medio de una consulta a la base de datos asociativa y tomarlo para construir el URL de la nueva solicitud Web. La segunda sección corresponde a la función PHP encargada de redireccionar la solicitud, la implementación de este componente se describe a continuación.

### **Redireccionamiento**

A raíz de la tabla presentada en la figura 51, la cual ilustra la comparación de algunas características descriptivas de interés para el caso de redireccionamiento de solicitudes Web. De acuerdo a las características del diseño presentadas en el capítulo anterior, se decidió utilizar el



modulo de redireccionamiento `mod_rewrite` de Apache, en conjunto con la función PHP `header()` para realizar la identificación, la reescritura y el redireccionamiento de las solicitudes Web que llegan al sistema de RI origen (Alejandría).

CARACTERÍSTICA	Alias	<code>mod_rewrite</code>	“refresh”	<code>header()</code>	JavaScript
Redireccionamiento externo	no	si	si	si	si
Redireccionamiento por página Web	si	si	si	si	si
Redireccionamiento grupal y por directorios	si	si	no	no	no
Ejecución del redireccionamiento	servidor	servidor	cliente	cliente	cliente
Acceso a una BD	no	si	no	no	no
Redireccionamiento de Robots Web	si	si	si	si	no
Capacidad para combinarse con un lenguaje de programación	no	si	si	si	si
Capacidad para redireccionar un URL codificado en un conjunto de caracteres determinado	no	no	no	si	si

Figura 51. Tabla comparativa de las características funcionales de las herramientas de direccionamiento (Alias, `mod_rewrite`, “refresh”, `header()` y JavaScript).

El servidor Web utilizará las directivas del `mod_rewrite` (Anexo A1): “`RewriteCond`” para evaluar el tipo de solicitud por medio de una expresión regular para denotar su forma y “`RewriteRule`” para dirigir la solicitud hacia la aplicación de reescritura y redireccionamiento correspondiente. Los desarrolladores del modulo de redireccionamiento (`mod_rewrite`) del servidor Apache (Bohen, 2006), sugieren que las aplicaciones que se ejecutan al activar una regla deben ser lo mas sencillas posible, evitando así problemas con el módulo y el rendimiento del servidor.

Las aplicaciones de redireccionamiento de solicitudes Web, luego de reescribir la solicitud de acuerdo al caso, utilizarán la función `header()` para redireccionar la nueva solicitud hacia el servidor Web destino. Como se muestra en la segunda sección del código descrito anteriormente, el URL correspondiente a la nueva solicitud (`<url_nuevo>`) se insertará en el parámetro “Location” de la función PHP mencionada, como sigue:

```
header("Location: <url_nuevo>");
```

### **Base de Datos**

Se utilizó el manejador de bases de datos (BBDD) PostgreSQL (versión 8.1) en la implementación de la base de datos necesaria para la asociación de los sistemas de repositorio. PostgreSQL es un sistema manejador de bases de datos de código abierto, basado en el enfoque objeto-relacional, con una estructura basada en el modelo cliente-servidor. Entre las razones por la que se utilizó este manejador de BBDD es que el sistema en estudio DSpace utiliza PostgreSQL para almacenar sus metadatos. Por esto resultó conveniente utilizar el mismo manejador de bases de datos para la BD asociativa; para facilitar las actividades del administrador del sistema, y adicionalmente contar con una base de datos 100% compatible con el repositorio destino, en especial al tomar en cuenta que algunos de los datos de la BD asociativa provienen de un vaciado a la base de datos del repositorio DSpace. Esto implica que entre otras cosas, se manejaría el mismo conjunto de caracteres para la codificación de los datos.

### **Archivo de configuración de casos de redireccionamiento**

Los casos de redireccionamiento se configurarán, de acuerdo al diseño presentado, en un archivo XML compuesto por las etiquetas descritas en el anexo A3 (Etiquetas de configuración de los casos de redireccionamiento). Cada etiqueta encapsula una variable que describe una característica del caso de redireccionamiento, un conjunto de etiquetas conforman la descripción completa de un mismo caso, mientras que un conjunto de casos debidamente etiquetados conforman el archivo XML. Como se definió en la etapa de diseño, este archivo es una de las entradas necesarias para la generación de las aplicaciones de redireccionamiento de solicitudes Web. Se decidió implementar estos detalles de la configuración en un archivo XML por la sencillez que este tipo de archivos suele aportar para los usuarios, lo cual se evidencia en la tendencia actual de este tipo de formato en los archivos de configuración de muchas herramientas. El lenguaje de las etiquetas utilizado en este archivo no es un estándar y no tiene definido un esquema para su validación, pero se puede verificar el significado y uso de dichas etiquetas en el anexo A3.

### **Interfaz gráfica de usuario**

Como se explicó en la sección 4.2, la configuración del conversor y generación de aplicaciones se maneja a través de una sencilla interfaz Web codificada en los lenguajes HTML y PHP, que será utilizada por el administrador del sistema conversor para su instalación, configuración y uso. Como se muestra en la figura 52, la interfaz se conforma por dos marcos: el marco izquierdo se compone de tres secciones:

- **Ayuda:** contiene los enlaces de ayuda para la instalación, configuración y uso del sistema.
- **Manejo de la DB asociativa:** Formada por los enlaces de creación, copiado y eliminación de tablas en la BD, así como un enlace con las instrucciones para crear la BD de PostgreSQL.
- **Generación de Archivos y Reglas:** La sección la conforman los enlaces que manejan el módulo de generación de archivos y reglas de redireccionamiento.

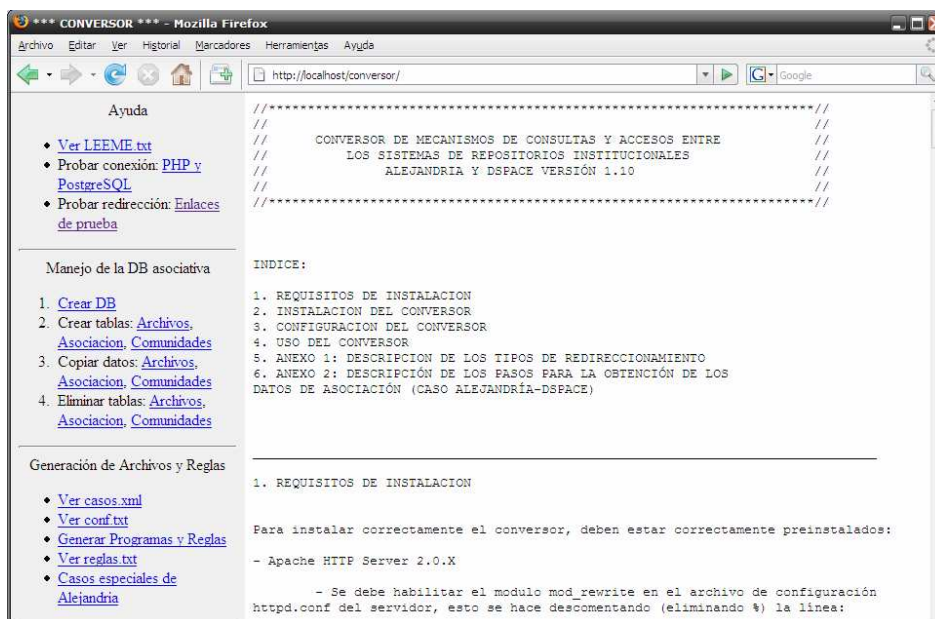


Figura 52. Interfaz gráfica del conversor.

Estos enlaces activan las acciones que generan la salida del sistema, cada acción produce un resultado que se muestra en el marco derecho. Este marco funciona solo como objetivo de las acciones seleccionadas en los enlaces del marco izquierdo, excepto en el caso de los enlaces de prueba; en que el marco derecho pasa a contener los enlaces de una página de pruebas desarrollada para el caso de estudio.

### Manual de administración del sistema

Uno de los objetivos planteados del proyecto fue desarrollar un manual de usuario destinado a facilitar la implantación, la configuración, el mantenimiento y la posible modificación del sistema. En base a este objetivo se desarrolló un documento con las instrucciones de instalación, configuración y uso del sistema. Este manual se presenta en detalle en el anexo A4 (Manual de

administración del sistema), aun así, a continuación se presenta una descripción básica de las secciones del mismo:

- **Requisitos de instalación:** Esta sección del manual define los requisitos necesarios para la instalación del sistema conversor.
- **Instalación del conversor:** Contiene las instrucciones para la instalación del sistema, la disposición de los archivos y la definición y descripción de los directorios creados en el proceso de instalación.
- **Configuración del conversor:** La sección está conformada por un conjunto ordenado de pasos necesarios para la configuración del sistema.
- **Uso del conversor:** Describe las instrucciones de uso en cada uno de los módulos del sistema conversor (ayuda, manejo de la BD asociativa, generación de archivos y reglas).
- **Descripción de los tipos de redireccionamiento:** Es un anexo que contiene la descripción técnica, así como ejemplos de los tipos de redireccionamiento definidos en el diseño del sistema.
- **Descripción de los pasos para la obtención de los datos de asociación:** Este anexo esta compuesto por una serie de instrucciones para la obtención de los datos de entrada a la BD asociativa para el caso Alejandría-DSpace.

## 5.2 Aplicación del conversor al caso de estudio

Debido a que el proyecto se desarrolla enmarcado en el caso de estudio entre los sistemas de RRII Alejandría y DSpace, y cumpliendo con el objetivo planteado de implantar y probar el conversor para este caso de estudio, se procedió a instalar el sistema en el servidor Web SABER-ULA y probar su funcionamiento.

El conversor se instaló, cumpliendo con los requisitos especificados en el manual de administración del sistema (ver anexo A4), en un servidor Web espejo utilizado por la unidad de desarrollo del Centro de Teleinformación de la ULA para realizar las pruebas de migración del mencionado RI. Este servidor de pruebas contiene una copia de la base de datos del servidor original manejada por: el sistema Alejandría, el servidor Web Apache y el manejador de BBDD Sybase. Se encuentra conectado por medio de una red local a un computador que funge como servidor de pruebas del sistema DSpace, el cual contiene los datos migrados (hasta mayo del 2008) desde Alejandría, el servidor DSpace está instalado sobre la plataforma Debian y utiliza como soporte para su funcionamiento al servidor de aplicaciones Tomcat y el manejador de BBDD PostgreSQL. La figura 53 muestra el esquema descrito anteriormente.

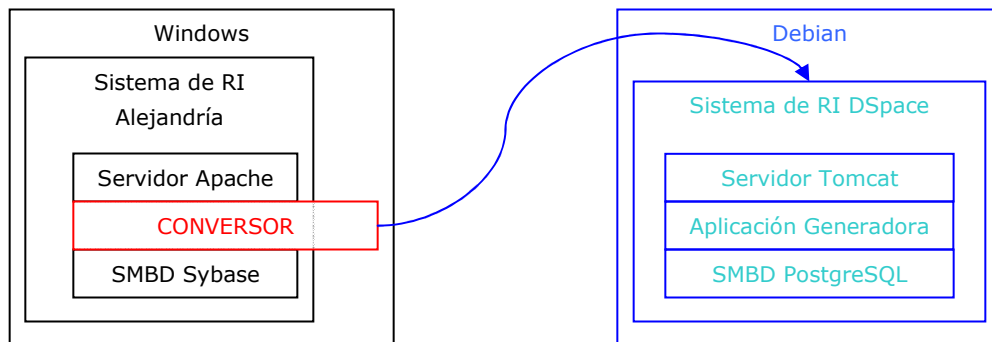


Figura 53. Esquema de la aplicación del convertor al caso de estudio.

La configuración del sistema convertor se realizó de acuerdo al caso de estudio:

- Se tomó a Alejandría como el sistema de RI origen y a DSpace como sistema destino.
- Se configuraron las variables de uso, entre ellas: el URL del repositorio origen y del repositorio destino, y las variables de conexión al manejador de BBDD (nombre de usuario, puerto, contraseña de ingreso).
- Se configuraron los casos mostrados en la figura 54, de acuerdo a las etiquetas de configuración de los casos de redireccionamiento (anexo A3), para la posterior generación de sus respectivas aplicaciones de redireccionamiento.

CASO	DESCRIPCIÓN
Autor	Búsqueda por autor.
Título	Búsqueda por título.
Descriptor	Búsqueda por descriptor.
FSIMPLE	Búsqueda por forma simple.
FGENERAL	Búsqueda por forma general.
DOC	Solicitud directa de documentos.
Acceso	Solicitud de registros de información por medio de la variable acceso.
Ejemplar	Solicitud de registros de información por medio de la variable ejemplar.
Documento	Solicitud de registros de información por medio de la variable documento.
Item	Solicitud de registros de información por medio de la variable item.
Codigo	Solicitud de registros de información por medio de la variable codigo.
POST	Solicitud de tipo POST.

Figura 54. Tabla de los casos identificados para el caso de estudio Alejandría-DSpace.

- Se creó la BD asociativa y se configuraron las tablas que contienen las características de asociación entre los sistemas Alejandría y DSpace (figura 55).

Tabla "asociación_id"	
id_origen	id_destino
Contiene el "acceso" a un registro de información en Alejandría	Contiene el "handle" correspondiente al registro de información en DSpace

Tabla "comunidad_id"	
comunidad_origen	id_destino
Contiene el nombre de una comunidad o grupo en Alejandría	Contiene el "handle" asociado a la comunidad correspondiente en DSpace.

Tabla "archivo_id"	
url_archivo	id_origen
Contiene la ruta a un documento en Alejandría.	Contiene el "acceso" correspondiente al documento en Alejandría.

Figura 55. Configuración de las tablas asociativas para el caso de estudio Alejandría-DSpace.

Posterior al proceso de configuración del conversor se ejecutó la generación de las aplicaciones y reglas de redireccionamiento, de esta acción resultaron doce aplicaciones codificadas en el lenguaje PHP e igual número de reglas mod\_rewrite (una por caso). Las reglas se copiaron al archivo de configuración del servidor Apache, el cual se encargará de dirigir las solicitudes de acuerdo a estas reglas hacia las correspondientes aplicaciones generadas; que fueron almacenadas en un directorio del conversor definido en el manual de administración (ver anexo A4).

### 5.3 Refinamiento por versiones y pruebas

El desarrollo del sistema se realizó basado en la metodología evolutiva denominada "modelo de versiones". El sistema evolucionó de manera incremental a medida que se desarrollaba una nueva versión, para cada una de estas versiones se realizaron pruebas de instalación y de redireccionamiento. A la versión final del sistema se le realizaron adicionalmente pruebas de

verificación de enlaces rotos y de rendimiento. La descripción de la evolución del sistema, así como las pruebas realizadas se presentan en las siguientes sub-secciones.

### **5.3.1 Versiones del sistema conversor**

Como ya se mencionó, el sistema se implementó de manera incremental, esta sección presenta una descripción de la evolución del mismo basada en las características funcionales de las versiones desarrolladas.

#### **Versiones 1 a 5**

Las primeras cinco versiones no contaban con un módulo automatizado de configuración y generación de aplicaciones y reglas, por el contrario, el sistema lo conformaba un conjunto de reglas y aplicaciones codificadas manualmente de acuerdo al caso de redireccionamiento. Los casos no se encontraban agrupados por tipos de solicitud, existía un caso para cada variable identificada, y muchas veces casos de un mismo tipo se codificaban con lógicas diferentes. No se había tomado en cuenta la codificación de los caracteres de las solicitudes Web, por lo que se produjeron fallas en las pruebas de redireccionamiento. Finalmente, el sistema evolucionó de la versión 1 a la 5, de acuerdo a la cantidad de casos de redireccionamiento identificados, codificados, instalados y probados en el caso de estudio.

#### **Versiones 6 a 9**

Durante esta etapa de desarrollo del sistema, se diseñó e implementó un módulo encargado de la configuración y la generación automatizada de las aplicaciones de redireccionamiento, así como la interfaz para el manejo del mismo. Para estas versiones aun no se había implementado el manejo de la BD que almacenaría las características asociativas de los sistemas en estudio, pero ya se encontraba diseñada la lógica de redireccionamiento de las solicitudes que harían uso de ella, también se diseñó la clasificación de las solicitudes Web utilizada por el generador de aplicaciones y reglas para la clasificación de los casos de redireccionamiento. En este punto ya se realizaban pruebas de configuración y generación del sistema, el cual evolucionó de acuerdo a la cantidad de casos generados automáticamente de manera exitosa.

## **Versiones 10 a 12**

Esta es la etapa final de desarrollo del sistema, aquí se implementó la base de datos asociativa y la generación automatizada de las aplicaciones que la utilizarían para redireccionar ciertos tipos de solicitudes. Se realizó el manual completo de administración del sistema y se mejoró la interfaz de configuración de manera que se pudieran realizar las acciones de creación, llenado y eliminación de las tablas asociativas, también se introdujeron enlaces de prueba de conexión al manejador de BBDD y de funcionamiento del lenguaje PHP dentro del servidor Web. Para este punto de la etapa de desarrollo del sistema, se realizaron pruebas de instalación, configuración, generación de aplicaciones y pruebas de rendimiento.

### **5.3.2 Pruebas**

Durante la etapa de implementación de este proyecto, el sistema fue sometido a diversas pruebas para garantizar su correcto funcionamiento. A continuación se presenta una síntesis de las pruebas aplicadas y los resultados obtenidos para el caso de estudio Alejandría-DSpace.

#### **Pruebas de instalación**

Cada versión finalizada fue instalada en diversos computadores que cumplieran con los requisitos establecidos en el manual de administración del sistema (ver anexo A4). La máquina de pruebas tenía instalados los dos servidores (Apache y Tomcat) utilizados en el caso de estudio simultáneamente, soportados por el sistema operativo Windows. Mientras que el servidor Apache respondía a las solicitudes enviadas al puerto 80, Tomcat respondía a las solicitudes enviadas al puerto 8080. La versión final del servidor se instaló en un servidor de pruebas de migración (como ya se explicó en la sección 5.2), que solo respondía a las peticiones realizadas a Alejandría a través del servidor Web Apache (puerto 80). Como resultado, se encontró que el conversor pasó todas las pruebas realizadas cuando se cumplieron los requisitos de instalación especificados.

#### **Pruebas de redireccionamiento**

Para este tipo de pruebas se utilizaron las herramientas de validación de casos de uso basadas en acciones visibles del usuario y las salidas esperadas del sistema (Sommerville, 2002): pruebas de caja negra y pruebas de caja blanca. Para esto, en primer lugar se construyó una página de pruebas (figura 57) con los siguientes casos de prueba: redireccionar accesos, redireccionar consultas y redireccionar documentos. Todos estos casos fueron generados de los casos de uso



descritos en la sección 4.2.1. La página esta compuesta por ejemplos de todos los casos de redireccionamiento identificados para el caso de estudio (figura 56), separados de acuerdo a los casos de prueba ya mencionados.

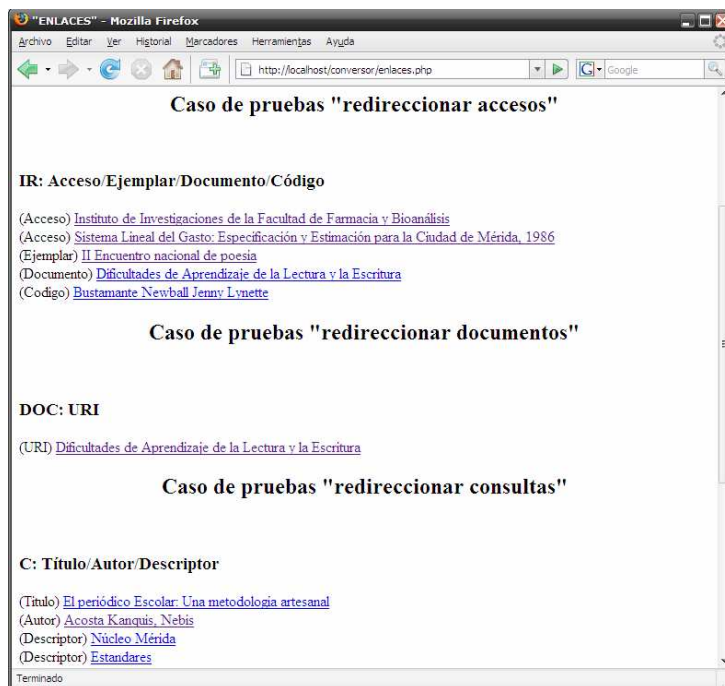


Figura 56. Página de pruebas de redireccionamiento para el caso de estudio Alejandría-DSpace.

La página mostrada en la figura anterior, para el caso en el que el método de envío es GET, contiene enlaces al servidor Web que soporta al sistema Alejandría. Cada enlace acciona una petición al repositorio compuesta por variables previamente identificadas como “redireccionables”. Además, la página contiene algunas cajas de texto utilizadas por el servidor SABER-ULA para realizar búsquedas enviadas mediante el método POST.

Las pruebas de caja negra constaron de la selección de los enlaces en la página de pruebas y los encontrados al navegar por el servidor, y la verificación de los casos de uso definidos en el capítulo 4 de acuerdo a la salida generada por cada acción. La figura 57 muestra algunos resultados obtenidos para diferentes casos de prueba con método de envío GET.

Redireccionamiento de consultas	
Inicial	http://www.saber.ula.ve/cgi-win/be_alex.exe?Autor=Acosta+Kanquis,+Nebis
Final	http://dspace.saber.ula.ve:8080/dspace/simple-search?query=((author:Acosta+Kanquis,+Nebis))
Redireccionamiento de accesos	
Inicial	http://www.saber.ula.ve/cgi-win/be_alex.exe?Acceso=T016300000223/0
Final	http://dspace.saber.ula.ve:8080/dspace/handle/123456789/8
Redireccionamiento de documentos	
Inicial	http://www.saber.ula.ve/db/ssaber/Edocs/pubelectronicas/educere/vol4num11/articulo4-11-1.pdf
Final	http://dspace.saber.ula.ve:8080/dspace/handle/123456789/21

Figura 57. Tabla de resultados de las pruebas de caja negra al conversor.

La figura 58 ilustra un resultado de una prueba de caja negra realizada a una caja de búsqueda con método de envío POST.



Figura 58. Resultado de una prueba de caja negra a una caja de búsqueda.

Las pruebas de caja blanca se realizaron accionando las solicitudes al servidor SABER-ULA y verificando la secuencia de pasos realizada por la aplicación de redireccionamiento a través de salidas de texto en formato HTML. La siguiente figura muestra el resultado obtenido para la solicitud:

[http://www.saber.ula.ve/cgi-win/be\\_alex.exe?Titulo=%22Proyecto+Sucre%22.+%BFUna+pol%EDtica+de+estado+para+la+frontera+venezolana](http://www.saber.ula.ve/cgi-win/be_alex.exe?Titulo=%22Proyecto+Sucre%22.+%BFUna+pol%EDtica+de+estado+para+la+frontera+venezolana)

```
El URL entró a la aplicación de redireccionamiento
query string del URL: Titulo=%22Proyecto+Sucre%22.+%BFUna+pol%EDtica+de+estado+para+la+frontera+venezolana
query string decodificado: Titulo="Proyecto+Sucre".+¿Una+política+de+estado+para+la+frontera+venezolana
query string codificado en el nuevo conjunto de caracteres: Titulo='Proyecto+Sucre'.+Â¿Una+polÃtica+de+estado+para+la+frontera+venezolana
URL Finál:
http://dspace.saber.ula.ve:8080/dspace/simple-search?query=((title:'Proyecto+Sucre'.+Â¿Una+polÃtica+de+estado+para+la+frontera+venezolana))
```

Figura 59. Resultado de una prueba de caja blanca aplicada al conversor.

En ambos tipos de prueba se consideró exitoso el redireccionamiento que realiza el conversor a las solicitudes Web.

### Pruebas de verificación de enlaces rotos

A través de la herramienta de verificación de enlaces rotos Fast Link Checker<sup>37</sup> (versión 1.6), se realizó una prueba al servidor Web SABER-ULA. La herramienta trabaja como un robot de una máquina de búsqueda (Googlebot, MSNBot, WISENubot, Alexa, etc.), ya que navega por todas las páginas de un sitio Web verificando todos sus enlaces (internos y externos). Al finalizar la verificación, la herramienta genera un reporte que contiene el total de enlaces encontrados, los procesados y los enlaces rotos hallados en el sitio.

La prueba se realizó en dos escenarios distintos al siguiente URL:

<http://www.saber.ula.ve>

Lo que significa que la herramienta se encargó de verificar todos los enlaces de las páginas estáticas pertenecientes al servidor Web SABER-ULA. El primer escenario consistió en verificar los enlaces rotos en el servidor cuando el conversor no estaba instalado, de allí se generó un reporte con los resultados mostrados en la figura 60.

<sup>37</sup> Fast Link Checker <http://www.webtweaktools.net/>

```
Checked site: http://www.saber.ula.ve/  
Total Links: 16382  
Processed Links: 16380  
Broken Links: 216  
Unique Links: 5517  
Total Hyperlinks: 4683  
Total Images: 6685  
Total Other: 5014  
Total Internal Links: 13011  
Total External Links: 3371
```

Figura 60. Resultado de la verificación de enlaces rotos al servidor SABER-ULA con el conversor desinstalado.

De los resultados obtenidos es importante tomar en cuenta tanto el total de enlaces en el sitio (Total Links: 16382) como los enlaces rotos (Broken Links: 216), lo que resulta en un 1.32% de enlaces rotos.

Seguidamente se procedió a instalar el conversor en el servidor Web y realizar la misma prueba descrita anteriormente, con el propósito de comparar la cantidad de enlaces rotos en ambos escenarios. El resultado de esta prueba se muestra a continuación en la figura 61.

```
Checked site: http://www.saber.ula.ve/  
Total Links: 16382  
Processed Links: 16380  
Broken Links: 166  
Unique Links: 5517  
Total Hyperlinks: 4683  
Total Images: 6685  
Total Other: 5014  
Total Internal Links: 13011  
Total External Links: 3371
```

Figura 61. Resultado de la verificación de enlaces rotos al servidor SABER-ULA con el conversor instalado.

En este caso, de un total de 16382 enlaces encontrados; 166 resultaron rotos, lo que resulta en un 1% de enlaces rotos aproximadamente. Comparando las cifras resultantes en ambos escenarios, existe una diferencia de 0.32% (50 enlaces) a favor del segundo escenario, es decir, hay menos enlaces rotos con el conversor instalado. Esto se debe a que cuando Alejandría encuentra un enlace roto, devuelve un error, mientras que cuando el sistema conversor está funcionando; redirecciona todos los enlaces encontrados en el servidor, en caso de que el registro no esté migrado aún a DSpace o el enlace esté roto en Alejandría, el sistema responde enviando al cliente a una página por omisión en el nuevo sistema de RI.

Aun así, se puede concluir que el conversor redirecciona al nuevo sistema de RI el 100% de las solicitudes que éste procesa a través de sus reglas y aplicaciones de reescritura y redireccionamiento.

### Pruebas de rendimiento

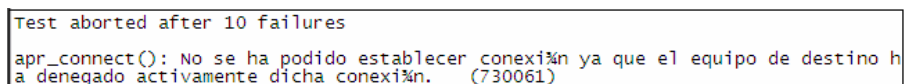
Se utilizó la herramienta de pruebas comparativas AB (Apache Benchmarking) del servidor Web Apache (Apache.org, 2008) para probar el rendimiento del servidor. El *benchmark* se aplicó con un total de mil (1000) peticiones en diferentes escenarios para el URL:

```
http://www.saber.ula.ve/cgi-win/be_alex.exe?Autor=Acosta+Kanquis,+Nebis
```

La sintaxis de la ejecución de esta prueba en la consola de trabajo de Windows es la siguiente:

```
C:\Archivos de programa\Apache Group\Apache2\bin>ab -n1000  
http://www.saber.ula.ve/cgi-win/be_alex.exe?Autor=Acosta+Kanquis,+Nebis
```

En primer lugar se aplicó la prueba con el servidor Apache detenido, el resultado se muestra en la siguiente figura.



```
Test aborted after 10 failures  
apr_connect(): No se ha podido establecer conexión ya que el equipo de destino h  
a denegado activamente dicha conexión. (730061)
```

Figura 62. Resultado del benchmarking al servidor Apache detenido.

Luego se realizó la misma prueba para el caso en que el servidor Apache se encontraba iniciado, el conversor desinstalado y el manejador de BBDD de Alejandría no estaba funcionando (figura 63). Lo que resultó en unos noventa minutos de duración de la prueba (*time taken for tests = 5364.15625 seg.*), y una media de 5364.016 ms por petición (*time per request*) debido al tiempo que se toma la aplicación CGI de Alejandría en darse cuenta que el manejador Sybase no estaba en funcionamiento para cada solicitud.

```

Server Software:      Apache/2.0.55
Server Hostname:     localhost
Server Port:         80

Document Path:       /cgi-win/be_alex.exe?Autor=Acosta+Kanquis,+Nebis
Document Length:     12516 bytes

Concurrency Level:    1
Time taken for tests: 5364.15625 seconds
Complete requests:   1000
Failed requests:     0
Write errors:        0
Total transferred:   12657046 bytes
HTML transferred:    12516000 bytes
Requests per second: 0.19 [#/sec] (mean)
Time per request:    5364.016 [ms] (mean)
Time per request:    5364.016 [ms] (mean, across all concurrent requests)
Transfer rate:       2.30 [kbytes/sec] received
    
```

Figura 63. Resultado del *benchmarking* al servidor Apache iniciado y manejador de BBDD detenido.

Al comparar esta última prueba con un escenario muy parecido al anterior, pero diferenciado debido a que en este caso el manejador de BBDD Sybase se encontraba en completo funcionamiento, se observa la reducción en la duración de la prueba (417.781250 seg.) a unos siete minutos (figura 64). Esto indica que el servidor tiene un mejor rendimiento cuando el manejador de BBDD se encuentra funcionando, y el tiempo por solicitud (*time per request*) observado es muy parecido al real, debido a que la base de datos utilizada para la prueba tenía un tamaño considerable.

```

Server Software:      Apache/2.0.55
Server Hostname:     localhost
Server Port:         80

Document Path:       /cgi-win/be_alex.exe?Autor=Acosta+Kanquis,+Nebis
Document Length:     12746 bytes

Concurrency Level:    1
Time taken for tests: 417.781250 seconds
Complete requests:   1000
Failed requests:     0
Write errors:        0
Total transferred:   12887000 bytes
HTML transferred:    12746000 bytes
Requests per second: 2.39 [#/sec] (mean)
Time per request:    417.781 [ms] (mean)
Time per request:    417.781 [ms] (mean, across all concurrent requests)
Transfer rate:       30.12 [kbytes/sec] received
    
```

Figura 64. Resultado del *benchmarking* al servidor Apache iniciado y manejador de BBDD funcionando.

Seguidamente se aplicó el *benchmark* para el caso en que el servidor Apache se encontraba corriendo, el convertor instalado y el servidor Tomcat estaba detenido (figura 65). Donde encontramos una media de 14.563 ms. en el tiempo de respuesta por petición, lo que indica que el convertor consume cantidades de tiempo muy pequeñas para realizar todas las actividades requeridas para reescribir y redireccionar una solicitud Web.

```

Server Software:      Apache/2.0.55
Server Hostname:     localhost
Server Port:         80

Document Path:       /cgi-win/be_alex.exe?Autor=Acosta+Kanquis,+Nebis
Document Length:     342 bytes

Concurrency Level:   1
Time taken for tests: 14.562500 seconds
Complete requests:   1000
Failed requests:     0
Write errors:        0
Total transferred:   529000 bytes
HTML transferred:    342000 bytes
Requests per second: 68.67 [#/sec] (mean)
Time per request:    14.563 [ms] (mean)
Time per request:    14.563 [ms] (mean, across all concurrent requests)
Transfer rate:       35.43 [Kbytes/sec] received
    
```

Figura 65. Resultado del *benchmarking* al servidor Apache iniciado, convertor instalado y servidor Tomcat detenido.

Al comparar el resultado obtenido, con el de la figura 66; caso en que el servidor Apache estaba encendido, el convertor instalado y el servidor Tomcat se encontraba corriendo, podemos notar una diferencia en el tiempo de duración de la prueba y en los tiempos por solicitud, esto se debe al tiempo que utiliza DSpace para consultar en su base de datos la información solicitada, en este caso esta BD contenía el 85% de los datos migrados desde Alejandria.

```

Server Software:      Apache/2.0.55
Server Hostname:     localhost
Server Port:         80

Document Path:       /cgi-win/be_alex.exe?Autor=Acosta+Kanquis,+Nebis
Document Length:     342 bytes

Concurrency Level:   1
Time taken for tests: 14.656250 seconds
Complete requests:   1000
Failed requests:     0
Write errors:        0
Total transferred:   529000 bytes
HTML transferred:    342000 bytes
Requests per second: 68.23 [#/sec] (mean)
Time per request:    14.656 [ms] (mean)
Time per request:    14.656 [ms] (mean, across all concurrent requests)
Transfer rate:       35.21 [Kbytes/sec] received
    
```

Figura 66. Resultado del *benchmarking* al servidor Apache iniciado, convertor instalado y servidor Tomcat iniciado.

Se puede concluir en principio que el repositorio institucional DSpace es bastante eficiente en comparación con Alejandria. La combinación del sistema operativo (Linux), la plataforma del RI (DSpace) y el manejador de base de datos (PostgreSQL) marcan la diferencia y se evidencia en una mejora del 3000% de la velocidad de respuesta del repositorio institucional con Dspace. Evidentemente, lo importante es que el tiempo de overhead del convertor no afecta el rendimiento de DSpace como nuevo servidor de solicitudes. Aunque, el convertor es un sistema que captura, procesa, reescribe y redirecciona cada una de las

solicitudes Web, el tiempo de este procesamiento es de 14 ms. Por tanto puede concluirse que su funcionamiento no afecta el servicio Web.



## Conclusiones

Los sistemas de repositorios institucionales proveen a los miembros de una determinada comunidad una herramienta invaluable para el almacenamiento, la gestión y la diseminación de contenido digital académico. Particularmente, el repositorio institucional (RI) SABER-ULA de la Universidad de Los Andes representa un medio para la preservación del patrimonio intelectual y su difusión al mundo entero a través de Internet.

El proceso de migración de un repositorio institucional perjudica notablemente su visibilidad Web. El portal SABER-ULA, luego del proceso de migración hacia la plataforma de Software Libre DSpace, pudiera perder más de diez mil enlaces (de los cuales aproximadamente la mitad son enlaces externos). La necesidad de un mecanismo que permitiera conservar los enlaces internos y externos al RI, sin modificación manual, inició y motivó el desarrollo de éste proyecto.

La implementación del sistema se basó en la metodología de enfoque evolutivo de desarrollo de software denominada “modelo de versiones”. Esta metodología fue de gran utilidad para el proyecto ya que no se encontraron antecedentes claros o evidentes donde se sugiriera el uso de una metodología específica de desarrollo de software para este tipo de problema. Los sistemas de repositorio debieron ser analizados y comparados antes de tener una idea básica del diseño del conversor. En particular el sistema de Alejandría carece de documentación a nivel de usuario y administración y esto dificultó su análisis. Por otro lado, tampoco había conocimiento previo de las herramientas disponibles para la implementación del conversor y de su utilidad. Por todas estas razones el desarrollo incremental ayudó a probar estas diversas herramientas en cada versión, permitiendo así modificar el diseño del sistema dependiendo de su utilidad y efectividad.

El análisis necesario para llevar a cabo el proyecto abarcó los sistemas de consulta y direccionamiento de documentos de los sistemas de RI en estudio. A raíz de la comparación de sus principales características se puede concluir que existen marcadas diferencias entre los sistemas en estudio, desde las plataformas hasta los estándares empleados en el manejo de los datos y sus interfaces. Dentro de estas diferencias se destaca, el conjunto de caracteres de cada sistema como una de las mayores dificultades para la reescritura de las peticiones Web, lo cual es una implicación propia de los procesos de migración de datos entre sistemas. Cabe resaltar que el conjunto utilizado por el sistema DSpace es el conjunto considerado estándar por las herramientas de intercambio y la iniciativa de Acceso Abierto.

Esta tesis presenta como valor agregado dos resultados: una interfaz de administración y un manual de usuario. La interfaz de administración del sistema tiene el propósito de facilitar las tareas del administrador, permitiendo la automatización de la configuración del sistema y la generación de las aplicaciones y reglas de redireccionamiento, evitando así los errores y la realización de acciones repetitivas. Por otra parte, el manual documenta los procesos de instalación, configuración, uso y posible modificación del conversor. Este manual incluye además, procedimientos específicos para el caso Alejandría-DSpace, necesarios para la configuración del sistema y generación de reglas y aplicaciones. La elaboración del manual necesitó un proceso incremental conjuntamente con el administrador del sistema, quien utilizó sus instrucciones en diversas oportunidades identificando fallas y deficiencias generando las diferentes versiones. De esta experiencia se puede concluir que la importancia de un manual radica; tanto en su correcta redacción, como en considerar e involucrar al usuario en su elaboración y finalmente en la comprensión por parte del usuario final, en este caso el Webmaster del RI.

Por otra parte, para la evaluación se realizaron diversas pruebas con el propósito de comprobar la funcionalidad y el rendimiento del sistema bajo un escenario de uso intenso. De los resultados obtenidos se muestra un buen rendimiento ya que el promedio de duración de la reescritura y redireccionamiento de una solicitud es de 14.656 ms. En total, el sistema tarda un promedio de 14 milisegundos en dar una respuesta al usuario, lo que mejora los tiempos de respuesta en un 3000% comparado con el sistema de RI origen (Alejandría). Por tanto, se puede decir que la solución propuesta del conversor no generará un problema de rendimiento o cuello de botella para el futuro servicio de RI.

El conversor propuesto pretende solucionar la mayor parte del problema de la visibilidad Web para el caso del repositorio SABER-ULA. Se puede estimar que se redireccionará aproximadamente un 90% de las solicitudes entrantes al servidor original y se puede concluir que la visibilidad se mantendría a un nivel aceptable. Además, el proceso de migración de la plataforma no afectará al usuario; ya que se enviará automáticamente las solicitudes del sistema origen al destino. Se deja en la mesa el estudio las estadísticas de funcionamiento del sistema y la revisión de sus deficiencias al momento de su puesta en marcha definitiva, para una posible mejora al sistema.

Se recomienda, luego de la migración definitiva identificar y automatizar los casos de redireccionamiento que no se pudieron tratar en este proyecto. Además para el caso de estudio, se pueden utilizar los conocimientos de un administrador del sistema DSpace, para identificar nuevas características funcionales de los sistemas de consulta y direccionamiento de documentos. Esto con el fin de realizar una completa y mejor adecuación de las aplicaciones de

reescritura y redireccionamiento de solicitudes Web al sistema destino. En este sentido se sugiere para el caso de estudio Alejandría-DSpace, tratar de lograr una correcta adecuación de la variable que describe un tipo de documento (Tiposdoc) en el sistema Alejandría, con su correspondiente metadato en el sistema DSpace. Esto se debe a que actualmente se pierde la posibilidad de filtrar las consultas adecuadamente cuando se presenta este caso, ya que en DSpace no existe una interfaz de consulta para éste metadato.

Además, dentro del proceso de configuración del conversor se sugiere realizar una validación previa del archivo de configuración de casos de redireccionamiento codificado en XML. Esto puede realizarse a través de un esquema (XML Schemas o DTD) y un parser, con el fin de evitar errores de lógica al momento de describir los casos de redireccionamiento y como consecuencia posibles errores en la sintaxis de las aplicaciones y reglas. Actualmente se asume que el Webmaster del repositorio tiene suficiente conciencia de los cambios para este archivo de configuración.

Para garantizar el acceso abierto al conocimiento, es necesaria la interfaz de recolección de metadatos OAI. Esta interfaz no fue contemplada en este proyecto ya que el equipo de migración de SABER-ULA aun no ha cubierto estos detalles. Sin embargo, se considera de vital importancia el caso de redireccionamiento de esta interfaz, la construcción de su correspondiente regla y/o aplicación de procesamiento de la solicitud. Esta interfaz debe ser analizada e incorporada al conversor a futuro para garantizar que los recolectores de metadatos en los cuales esta suscrito SABER-ULA puedan seguir realizando sus tareas de recolección normalmente.

Para finalizar, cabe mencionar que este proyecto deja un aporte al estudio de las herramientas de redireccionamiento existentes, analizadas desde un punto de vista utilitario y las combina adecuadamente con la finalidad de optimizar el proceso y adecuarlo al caso de estudio.

## Anexos

### A1. Directivas del módulo `mod_rewrite` de Apache

A continuación se presenta el conjunto de directivas de reescritura de URL pertenecientes al módulo de redireccionamiento `mod_rewrite` del servidor Web Apache:

1. **RewriteBase:** Esta directiva configura explícitamente el URL base para las reescrituras por directorio. Es decir, actúa localmente, cortando la sección correspondiente al directorio local del URL mientras este es procesado, aplica las reglas de reescritura solo en la sección restante, y al final del procesamiento añade de nuevo el prefijo del directorio a la ruta para reenviar el URL al procesamiento de rutas del servidor Web.
2. **RewriteCond:** Esta directiva define una regla o condición de reescritura. De manera que se usa la siguiente directiva, solo si el URI cumple la condición descrita. Entre las condicionales a comparar, encontramos:
  - Cabeceras HTTP
  - Variables internas del servidor.
  - Variables de conexión y petición.
  - Variables del sistema
  - Otras variables especiales.

Adicionalmente estas reglas o condiciones pueden ser configuradas por medio de banderas, entre las cuales tenemos:

- `'nocase (NC)`: lo que hace que la evaluación no sea sensible a mayúsculas.
  - `'ornext (OR)`: se usa para combinar las condiciones de la regla con un operador OR local en vez del operador AND implícito por omisión.
3. **RewriteEngine:** Activa o desactiva la máquina de reescritura del `mod_rewrite`. Si se configura desactivado (`off`), el módulo no realiza ningún procesamiento. Por el contrario, al activarlo (`on`), el módulo es capaz de realizar las operaciones de reescritura y redireccionamiento de URL para las que fue diseñado.
  4. **RewriteLock:** Esta directiva configura el nombre de un archivo cerrado de sincronización necesario para que el `mod_rewrite` se comuniqué con programas de mapeo de la directiva `RewriteMap`.

5. **RewriteLog**: Esta directiva configura el nombre de un archivo al cual el servidor ingresa cualquier acción de reescritura que realiza.
6. **RewriteLogLevel**: Esta directiva configura el nivel de verbosidad del archivo configurado en la directiva **RewriteLog**.
7. **RewriteMap**: Define un mapa de reescritura que puede ser usado dentro de las cadenas de reglas de sustitución, por las funciones de mapeo para insertar o sustituir campos del URL a través de una búsqueda por clave. Cuya fuente puede ser de diferentes tipos:
  - **Texto plano (txt)**: el mapa es un archivo ASCII plano que contiene separado por espacios en blanco los pares asociativos necesarios para el mapeo.
  - **Texto plano aleatorizado (rnd)**: Es muy similar al mapa de texto plano, solo que adicional a la búsqueda del valor en el mapa, se selecciona aleatoriamente una de las posibles alternativas de respuesta, si las hay, que vienen separadas por el caracter | (OR).
  - **Archivo hash (dbm)**: En este caso la fuente es un archivo en formato binario con el mismo contenido que en el caso del mapa de texto plano, pero en una representación especial que optimiza la duración de las búsquedas.
  - **Funciones internas (int)**: La fuente es una función interna de Apache, entre las que encontramos:
    - **Toupper**: convierte las palabras clave a mayúsculas.
    - **Tolower**: Convierte las palabras clave a minúsculas
    - **Escape**: Traduce los caracteres especiales de las palabras clave a código hexadecimal.
    - **Unescape**: Traduce el código hexadecimal contenido en las palabras clave, de nuevo en caracteres especiales.
  - **Programas externos de reescritura (prg)** en este caso la fuente es un programa codificado en cualquier lenguaje de programación, cuyo resultado debe ser un programa ejecutable cuya primera línea es la ruta al compilador.
8. **RewriteRule**: Esta directiva es la que realmente realiza el trabajo de reescritura. Aplicando al URL una expresión regular que lo reescribe de acuerdo a la misma. Adicionalmente la regla puede ser configurada por medio de banderas, entre las cuales están:
  - **'chain (C)**: Encadena la regla actual con la siguiente
  - **'cookie (CO)**: Configura un cookie.
  - **'env (E)**: Configura una variable de ambiente
  - **'last (L)**: Detiene el proceso de reescritura de URL y no aplica más reglas.

- ‘next (N): Reinicia el proceso de reescritura desde la primera regla con el URL resultante de la aplicación de la regla actual.
- ‘nocase (NC): Hace que el patrón de la expresión regular evaluadora no sea sensible a las mayúsculas.

## A2. Glosario

**Acceso abierto:** Disponibilidad libre en Internet de literatura de carácter académico y científico, permitiendo determinados usos como descargar, copiar, imprimir y recuperar el texto completo de los documentos (Universidad Carlos III de Madrid, 2007).

**Auto-archivado:** Acto por el que el autor deposita su documento en un sitio Web públicamente accesible (Universidad Carlos III de Madrid, 2007).

**Apache:** Servidor Web de código abierto. Su desarrollo comenzó en febrero de 1995, por Rob McCool, en una tentativa de mejorar el servidor existente en el NCSA. La primera versión apareció en enero de 1996, el Apache 1.0. Hacia el 2000, el servidor Web Apache era el más extendido en el mundo. El nombre «Apache» es un acrónimo de «a patchy server» -un servidor de remiendos-, es decir un servidor construido con código preexistente y piezas y parches de código. Es la auténtica «kill app» del software libre en el ámbito de los servidores y el ejemplo de software libre de mayor éxito, por delante incluso del kernel Linux. Desde hace años, más del 60% de los servidores Web de Internet emplean Apache (Interactive Programmers Community, 2000).

**ASCII (American Standard Code for Information Interchange):** Un esquema estándar de codificación de caracteres utilizado por la mayoría de los computadores para visualizar letras, dígitos y caracteres especiales (Interactive Programmers Community, 2000).

**BOAI (Budapest. Iniciativa Acceso Abierto):** Iniciativa del movimiento de acceso libre al conocimiento, que resultó de una reunión celebrada en Budapest a finales de 2001, promovida por la Open Society Institute. Se promulgó una declaración en la que se establecen los principios acerca del significado y ámbito del acceso libre y se definen estrategias complementarias para promover dicho acceso (Universidad Carlos III de Madrid, 2007).

**CGI (Common Gateway Interface):** Interfaz Común de Pasarela. Interface de intercambio de datos estándar en WWW a través del cual se organiza el envío de recepción de datos entre visualizadores y programas residentes en servidores WWW (Interactive Programmers Community, 2000).

**Declaración de Berlín:** Declaración suscrita en octubre de 2003 por las más importantes instituciones científicas europeas, en apoyo del acceso abierto y de los archivos abiertos, y

animando a los investigadores a depositar sus trabajos en archivos abiertos (Universidad Carlos III de Madrid, 2007).

**Expresión Regular:** Una expresión regular es un patrón que puede concordar con diversas cadenas de texto (nongnu.org, 2007)

**HTML (HyperText Markup Language):** Lenguaje de marcado de Hipertexto. Es el lenguaje estándar para describir el contenido y la apariencia de las páginas en el WWW (Interactive Programmers Community, 2000).

**HTTP (Hiper Text Transfer Protocol):** Protocola de transferencia de Hipertexto. Es el protocolo de Internet que permite que los exploradores del WWW recuperen información de los servidores. Es un protocolo de aplicación con la sencillez y velocidad necesaria para sistemas de información distribuidos, colaborativos y de diferentes medios. Es un protocolo general, independiente y orientado a objetos usado para diferentes tareas, como sistemas de nombres de servidores y de administración de objetos distribuidos, a través de la extensión de sus métodos (comandos). Una característica de HTTP es la forma de representar los datos, permitiendo a los sistemas funcionar independientemente de los datos siendo transferidos. HTTP ha sido usado por el WWW desde 1990 (Interactive Programmers Community, 2000).

**Metadatos:** Los metadatos son datos asociados a un documento digital que recogen información fundamentalmente descriptiva (autor, título, etc.). También pueden incluir información de administración (creación del recurso, derechos, control de acceso, etc.), y preservación (tipo de formato, etc.) (Universidad Carlos III de Madrid, 2007).

**OAI (Open Archives Initiative):** Iniciativa surgida en 1999 con el objetivo de desarrollar y promover estándares de interoperabilidad para la difusión y recuperación de publicaciones científicas en Internet. Esta iniciativa surge en el seno de las comunidades de e-prints, como alternativa a la distribución de información científica en manos de los tradicionales editores comerciales. La interoperabilidad se favorece con el establecimiento del protocolo OAI-PMH, que facilita la comunicación de metadatos codificados en el formato Dublín Core, entre servidores proveedores y clientes recolectores (Universidad Carlos III de Madrid, 2007).

**PHP (Preprocessed Hypertext Pages):** PHP es un lenguaje de scripting embebido en HTML. Mucha de su sintaxis es tomada de C, Java y Perl con un par de características adicionales únicas



y específicas de PHP. El propósito del lenguaje es permitir que los desarrolladores Web escriban páginas generadas dinámicamente con rapidez (Interactive Programmers Community, 2000).

**Repositorio Institucional:** Es un sistema de información electrónico que sirve para almacenar, preservar y difundir la producción intelectual de una determinada institución, normalmente una comunidad universitaria (Universidad Carlos III de Madrid, 2007).

**Script:** Secuencia de código sin compilar que puede ejecutar acciones, usualmente poco complejas, mediante un software que la interpreta (Interactive Programmers Community, 2000).

**TCP/IP (Transmission Control Protocol/Internet Protocol):** Se trata de un estándar de comunicaciones muy extendido y de uso muy frecuente para software de red basado en Unix con protocolos Token-Ring y Ethernet, entre otros. Es compatible con productos de muchas marcas: IBM, DEC, Sun, AT&T, Data General, etc. TCP/IP es conforme a los niveles 3 y 4 de los modelos OSI. Este conjunto de protocolos fue desarrollado originalmente para el Departamento de Defensa de Estados Unidos (Interactive Programmers Community, 2000).

**URI (Uniform Resource Indicator):** Un Identificador Universal de Recursos es un miembro de este conjunto universal de nombres, en un cierto espacio de nombres y con una dirección referida a un cierto protocolo, ambos previamente registrados (Interactive Programmers Community, 2000).

**URL (Universal Resource Locator):** Localizador Universal de Recursos. Sistema unificado de identificación de recursos en la red. Las direcciones se componen de protocolo, FQDN y dirección local del documento dentro del servidor. Permite identificar objetos WWW, Gopher, FTP, News, etc. Una cadena que suministra la dirección Internet de un sitio Web o de un recurso World Wide Web, junto con el protocolo por el que se tiene acceso a ese sitio o a ese recurso. El tipo más común de dirección URL es http://, que proporciona la dirección Internet de una página Web (Interactive Programmers Community, 2000).

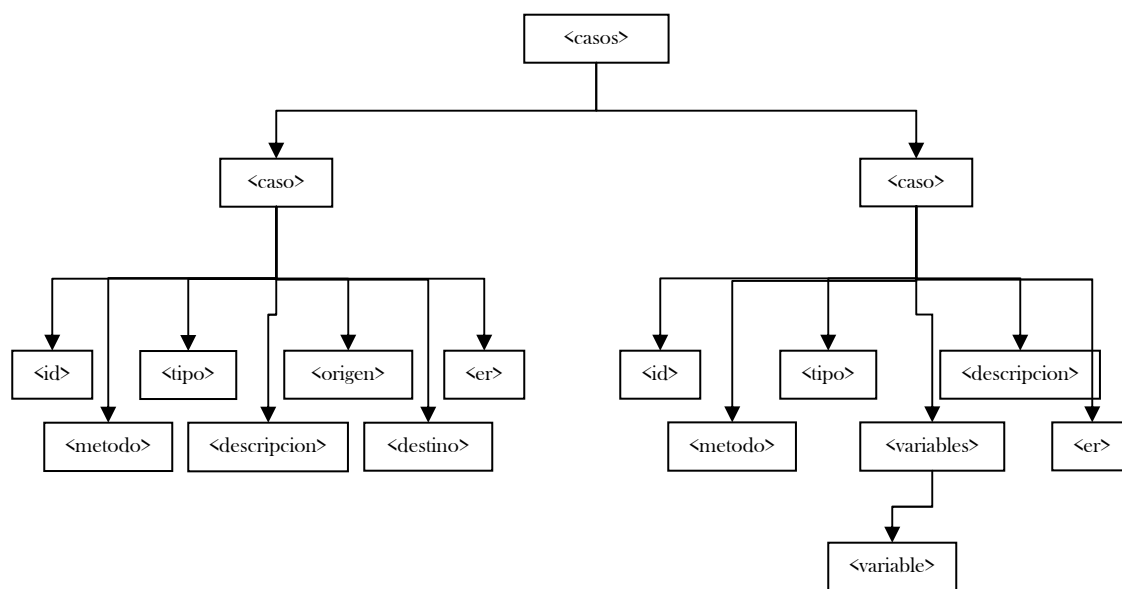
**Web:** Por éste término se suele conocer a WWW (World Wide Web), creado por el Centro Europeo de Investigación Nuclear como un sistema de intercambio de información y que Internet ha estandarizado. Supone un medio cómodo y elegante, basado en multimedia e hipertexto, para publicar información en la red. Inicial y básicamente se compone del protocolo

http y del lenguaje html. Un ejemplo de páginas de éste tipo, es la que tienes delante en estos momentos (Interactive Programmers Community, 2000).

**XML (Extensible Markup Language):** Es un meta-lenguaje que permite definir lenguajes de marcado adecuados a usos determinados. En la práctica corresponde a un estándar que permite a diferentes aplicaciones interactuar con facilidad a través de La Red. XML fue creado al amparo del World Wide Web Consortium (W3C) organismo que vela por el desarrollo de WWW partiendo de las amplias especificaciones de SGML. Su desarrollo se comenzó en 1996 y la primera versión salió a la luz el 10 de febrero de 1998. La primera definición que apareció fue: Sistema para definir validar y compartir formatos de documentos en la Web (Interactive Programmers Community, 2000).

### A3. Etiquetas de configuración de los casos de redireccionamiento

Este anexo presenta una descripción de las etiquetas utilizadas para configurar el archivo de casos de redireccionamiento necesario para la generación automatizada de las aplicaciones y reglas de redireccionamiento. La siguiente figura ilustra a través de un árbol, la disposición correcta de las etiquetas en el archivo XML de configuración de casos.



Modelo de objetos de documentos (DOM) del archivo casos.xml

1. <casos>

Es la etiqueta más general, agrupa todos los casos a generarse, por lo que dentro de ella deben estar todas las demás etiquetas.

2. <caso>

Contiene un caso y lo separa de otro, dentro de esta etiqueta están los elementos necesarios para identificar un caso de redireccionamiento.

3. <id>

Enumera de manera ordenada y creciente el caso al que corresponde la misma.

4. <metodo>

Cada caso está caracterizado por el método de envío de solicitudes, puede ser GET o POST.

5. <tipo>

Clasificación de los casos según el método y además las características de acceso, resultado y consulta. El tipo determina el tratamiento del redireccionamiento.

6. <descripcion>

Incluye una cadena de texto con una pequeña descripción del caso de redireccionamiento.

7. <origen>

La variable del Sistema de Repositorio origen (Alejandría) cuyo valor es de interés para el caso.

8. <destino>

La variable del Sistema de Repositorio destino (DSpace) asociada a la variable del origen (Alejandría) que va a contener el valor de interés proveniente de la misma.

9. <er>

Contiene la expresión regular utilizada en el servidor Apache para gestionar el redireccionamiento de acuerdo al caso. Indica como identificar o reconocer en los enlaces (URL) del sistema origen la ocurrencia del caso.

10. <variables>

Contiene las variables necesarias para realizar un redireccionamiento complejo de tipo POST o FORMA, que puede contener más de una variable de interés.

11. <variable>

Contiene el nombre de la variable del sistema de Repositorio origen (Alejandría) de interés para el redireccionamiento.

## A4. Manual de administración del sistema

A continuación se presenta una copia del manual de administración del sistema conversor de mecanismos de consultas y accesos entre los sistemas de repositorios institucionales Alejandría y DSpace.

```
/**
//
//      CONVERSOR DE MECANISMOS DE CONSULTAS Y ACCESOS ENTRE
//      LOS SISTEMAS DE REPOSITARIOS INSTITUCIONALES
//      ALEJANDRIA Y DSPACE VERSIÓN 1.12
//
//      MANUAL DE ADMINISTRACION DEL SISTEMA
//
//
//**
```

INDICE:

1. REQUISITOS DE INSTALACION
2. INSTALACION DEL CONVERSOR
3. CONFIGURACION DEL CONVERSOR
4. USO DEL CONVERSOR
5. ANEXO 1: DESCRIPCION DE LOS TIPOS DE REDIRECCIONAMIENTO
6. ANEXO 2: DESCRIPCIÓN DE LOS PASOS PARA LA OBTENCIÓN DE LOS DATOS DE ASOCIACIÓN (CASO ALEJANDRÍA-DSPACE)

---

### 1. REQUISITOS DE INSTALACION

Para instalar correctamente el conversor, deben estar correctamente preinstalados:

- Apache HTTP Server 2.0.X

- Se debe habilitar el modulo `mod_rewrite` en el archivo de configuración `httpd.conf` del servidor, esto se hace descomentando (eliminando `%`) la línea:

```
LoadModule rewrite_module modules/mod_rewrite.so
```

- Se debe activar el modulo `mod_rewrite` en el archivo de configuración `httpd.conf` del servidor, esto se hace agregando la siguiente línea luego de las directivas de creación del directorio virtual:

```
RewriteEngine on
```

- PHP 5

- Se debe habilitar la extensión pgconnect de PHP:

En windows, esto se hace descomentando la línea:

```
extension=php_pgsql.dll  
extension=php_socket.dll
```

en el archivo php.ini. Y se debe colocar la extensión php\_pgsql.dll en el mismo directorio del archivo php.ini

- PostgreSQL 8.1.X

- Se debe crear una base de datos con las siguientes características preferiblemente:

```
NOMBREBD = conversor  
PROPIETARIO = postgres  
ENCODING = 'SQL_ASCII'  
TABLESPACE = pg_default
```

Nota: Para mayor información al respecto haga click sobre el enlace "Crear DB" de la interfaz web del conversor, o abra el archivo "crear\_db.txt" que se encuentra en el directorio /tablas del conversor.

---

## 2. INSTALACION DEL CONVERSOR

- Descomprima el archivo en el directorio raíz de su servidor web:

Se creará un directorio llamado conversor que contiene todo lo necesario para su correcto funcionamiento.

La disposición de los archivos y subdirectorios es la siguiente:

/conversor

  /casos

    caso#.php

  /conf

    conf.txt  
    casos.xml  
    prueba.php  
    prueba.html

  /lib

```

caso.GET.C.txt
caso.GET.FORMA.txt
caso.GET.IR.txt
caso.GET.DOC.txt
caso.POST.txt
casos_especiales.txt

```

/servidor

```
reglas.txt
```

/tablas

```

archivos.php
crear_db.txt
crear_tablas.php
copiar_datos.php
eliminar_tablas.php
sql_alejandria.txt
sql_dspace.txt
archivotoacceso_saber.txt
archivotocodigo_saber.txt

```

Tabla de Directorios

NOMBRE DEL DIRECTORIO	ESPECIFICACION DEL DIRECTORIO
conversor	directorio raíz del conversor, contiene lo necesario para el funcionamiento del conversor.
casos	contiene los programas de redireccionamiento de URL's según el caso, producto de la ejecución del generador
conf	contiene los archivos de configuración necesarios para el funcionamiento del conversor. Un archivo conf que contiene las variables globales a usar, y el archivo casos.xml que contiene los casos a generar.
lib	contiene las plantillas que se usan para generar los programas de redireccionamiento automáticamente.
servidor	contiene los elementos que relacionan al conversor con el servidor web, como el archivo de reglas mod_rewrite
tablas	contiene los archivos y elementos necesarios para el manejo de las tablas necesarias para la asociación de algunos elementos del conversor.

### 3. CONFIGURACION DEL CONVERSOR

- Configure las variables globales del conversor:

Abra el archivo conf.txt del directorio /conf de su conversor y modifique de acuerdo a sus preferencias cada una de las variables allí dispuestas. Puede agregar todas las variables que desee, pero recuerde que este archivo será una cabecera para otros programas usados por el conversor, por lo que debe escribir correctamente cada una de las variables que agregue o modifique.

- Configure los casos que desea generar:

Abra el archivo casos.xml del directorio /conf de su conversor y agregue o modifique los casos que desea generar para el conversor. A continuación se explica cada una de las etiquetas del archivo:

<casos>

Es la etiqueta mas general, agrupa todos los casos a generarse, por lo que dentro de ella deben estar todas las demás etiquetas.

<caso>

Contiene un caso y lo separa de otro, dentro de esta etiqueta están los elementos necesarios para identificar un caso de redireccionamiento.

<id>

Enumera de manera ordenada y creciente el caso al que corresponde la misma.

<metodo>

Cada caso está caracterizado por el método de envío de URL's utilizado. puede ser GET o POST.

<tipo>

Clasificación de los casos según el método y además las características de acceso, resultado y consulta. El tipo determina el tratamiento del redireccionamiento:

	C	(consultas)
GET	FORM	(formularios)
	DOC	(documentos)
	IR	(identificadores de registros)
POST	POST	

<descripcion>

Incluye una cadena de texto con una pequeña descripción del caso de redireccionamiento.

<origen>

La variable del Sistema de Repositorio origen (Alejandría) cuyo valor es de interés



para el caso.

<destino>

La variable del Sistema de Repositorio destino (DSpace) asociada a la variable del origen (Alejandría) que va a contener el valor de interés proveniente de la misma.

<er>

Contiene la expresión regular utilizada en el servidor Apache para gestionar el redireccionamiento de acuerdo al caso. Indica como identificar o reconocer en los enlaces (URL) del sistema origen la ocurrencia del caso.

<variables>

Contiene las variables necesarias para realizar un redireccionamiento complejo de tipo POST o FORMA, que puede contener mas de una variable de interés.

<variable>

Contiene el nombre de la variable del sistema de Repositorio origen (Alejandría) de interés para el redireccionamiento.

Nota: Los casos se diferencian por la manera como deben ser tratados para redireccionar correctamente los URL's. Además deben separarse para que cada programa de redireccionamiento sea lo mas sencillo posible, evitando así problemas con el módulo de redireccionamiento del servidor Apache.

---

#### 4. USO DEL CONVERSOR

- Si no lo ha hecho, abra un navegador y corra el conversor desde el directorio raíz (DocumentRoot) o directorio virtual de su servidor:

```
<raiz>/conversor/index.html
```

A la izquierda de su navegador encontrará un marco con todos los enlaces que hacen uso del conversor.

\*..AYUDA..\*

- Al hacer click en el enlace "leeme" podrá visualizar este archivo en su navegador

- Pruebe la conexión de su servidor con PHP y PostgreSQL haciendo click en el enlace:

```
"Probar conexión: PHP y PostgreSQL"
```

Nota: La contraseña no debe contener en ningún lugar la frase "post", de ser así modifíquela en su manejador de BD antes de usar el conversor.

\*..MANEJO DE LA DB ASOCIATIVA..\*

- Si no lo ha hecho, haga click sobre el enlace "Crear DB" y siga los pasos descritos para la creación de la DB asociativa "conversor".
  
- Para crear las tablas asociativas (archivo\_id y asociacion\_id) haga click sobre el enlace:
  - Archivos: Para crear la tabla "archivo\_id" necesaria para la asociación entre la ruta de un documento y su identificador en el sistema de repositorio origen (correspondiente al acceso Alejandría).
  
  - Asociación: Para crear la tabla "asociacion\_id" necesaria para la asociación entre el identificador en el sistema de repositorio origen (acceso Alejandría) y su equivalente en el sistema destino (Handle de DSpace).
  
  - Comunidades: Para crear la tabla "comunidad\_id" necesaria para la asociación entre el nombre de la comunidad en el sistema de repositorio origen, y si correspondiente identificador en el sistema destino.
  
- Para copiar los datos a las tablas correspondientes haga click sobre el enlace:

- Archivos: Para copiar los datos del archivo doc\_acceso.sql a la tabla archivo\_id.

Nota: El archivo doc\_acceso.sql debe contener una lista con los datos que asocian la ruta de un documento y su correspondiente identificador en el sistema de repositorio origen (correspondiente al acceso Alejandría) ordenados de la siguiente manera:

RUTA\_AL\_DOCUMENTO.ext,ACCESO

Por ejemplo:

/ssaber/Edocs/centros\_investigacion/ciaal/agroalimentaria/anum1/inotas1\_1.pdf,T016300000002/8  
/ssaber/Edocs/Curriculum/arquitectura/hr\_debby-avendano.htm,U01630003041  
/ssaber/Edocs/Curriculum/farmacia/hr\_yajaira-maldonado.htm,U01630003042

IMPORTANTE: Ver Anexo 2 para generar el archivo doc\_acceso.sql.

- Asociación: Para copiar los datos del archivo acceso\_handle.sql a la tabla asociacion\_id

Nota: El archivo acceso\_handle.sql debe contener una lista con los datos que asocian el identificador en el sistema de repositorio origen (correspondiente al acceso Alejandría) de un documento y su equivalente en el sistema destino (Handle DSpace) ordenados de la siguiente manera:

ACCESO\_ALEJANDRÍA,HANDLE

Por Ejemplo:

T016300000767/0,5  
T016300003838/0,6  
T016300001358/0,17

IMPORTANTE: el archivo acceso\_handle.sql debe ser generado y ordenado por el administrador del sistema luego del proceso de migración, en el que debe haberse migrado el identificador en el sistema de repositorio origen (acceso Alejandría) del documento como un metadato del nuevo sistema (DSpace).

IMPORTANTE: Ver Anexo 2 para generar el archivo acceso\_handle.sql.

- Comunidades: Para copiar los datos del archivo comunidad\_handle.sql a la tabla comunidad\_id

Nota: el archivo comunidad\_handle.sql debe ser generado y ordenado por el administrador del sistema luego del proceso de migración, debe contener una lista con los datos que asocian el nombre de una comunidad en el sistema de repositorio origen y su correspondiente identificador en el sistema destino (Handle DSpace) ordenados de la siguiente manera:

NOMBRE\_COMUNIDAD, HANDLE

IMPORTANTE: Ver Anexo 2 para generar el archivo comunidad\_handle.sql.

MUY IMPORTANTE: Por cada vez que usted haga click sobre uno de los enlaces anteriores se añadirán al final de la tabla los datos que se encuentren en los archivos doc\_acceso.sql o acceso\_handle.sql sin importar que los datos ya esten alli.

- Para eliminar las tablas asociativas haga click sobre el enlace:

- Archivos: Para eliminar la tabla archivo\_id
- Asociacion: Para eliminar la tabla asociacion\_id
- Comunidades: Para eliminar la tabla comunidad\_id

IMPORTANTE: Al borrar las tablas borrará los datos almacenados en ellas.

Nota: Si desea modificar o agregar nuevos datos a las tablas, se aconseja eliminar las tablas actuales, volverlas a crear, actualizar los archivos de entrada doc\_acceso.sql o acceso\_handle.sql, segun sea el caso y volver a copiar los datos a la tabla.

\*..GENERACION DE ARCHIVOS Y REGLAS..\*

- Haga click sobre el enlace "Ver casos.xml" para visualizar en el navegador el archivo casos.xml que contiene los casos a ser generados debidamente

organizados y etiquetados.

- Haga click sobre el enlace "Ver conf.txt" para visualizar el archivo de configuración conf.txt, que contiene las variables globales a utilizar por los programas redireccionadores luego de ser generados. Si desea lo puede modificar.

- Haga click sobre el enlace "Generar Programas y Reglas" para generar en el directorio /casos, los programas redireccionadores correspondientes a los casos descritos en el archivo casos.xml. Además se generará en el directorio /servidor el archivo reglas.txt con las correspondientes reglas de redireccionamiento utilizadas por el servidor apache para redireccionar las solicitudes realizadas al servidor.

- Haga click sobre el enlace "Ver reglas.txt" para visualizar en el navegador el contenido del archivo reglas.txt.

MUY IMPORTANTE: El contenido de este archivo debe ser copiado en el archivo de configuración httpd.conf del servidor inmediatamente después de la línea:

```
RewriteEngine on
```

y antes que alguna otra.

- Haga click sobre el enlace "Casos especiales de Alejandria" para visualizar en el navegador el contenido del archivo casos\_especiales.txt. Observará algunas reglas especiales de redireccionamiento para el caso específico Alejandria-DSpace.

Nota: Si usted desea puede copiar el contenido de este archivo en el archivo de configuración httpd.conf del servidor junto a las demás reglas.

---

## 5. ANEXO 1: DESCRIPCIÓN DE LOS TIPOS DE REDIRECCIONAMIENTO

- C:

Se caracteriza porque el URL del sistema de repositorio origen contiene un Query String compuesto por variables que contienen palabras clave para la búsqueda de un documento.

El resultado de la acción es cero, uno o más registros que contienen las palabras clave contenidas en las variables del URL.

Su caso de redireccionamiento funciona tomando el valor de la variable del URL original, para luego construir un nuevo Query String adaptado al sistema de repositorio destino (DSpace) y finalmente redireccionarlo.

Ejemplos: consulta por descriptor, título, autor, palabras claves, etc.

- FORM:

Se caracteriza porque el URL del sistema de repositorio origen contiene un Query String iniciado con la variable "forma=" seguido del tipo de formulario utilizado para el envío de valores, y variables que contienen las palabras clave asociadas a la búsqueda.

El resultado de la acción es cero, uno o más registros que contienen las palabras clave contenidas en las variables del URL asociadas a la búsqueda.

Su caso de redireccionamiento funciona tomando el valor de las variables del URL original, para luego construir un nuevo Query String adaptado al sistema de repositorio destino (DSpace) y finalmente redireccionarlo.

Ejemplos: forma general, forma simple de Alejandría con una o más variables de búsqueda

- DOC:

Se caracteriza porque el URL del sistema de repositorio origen no contiene Query String, contiene una ruta directa a un documento perteneciente al servidor (URI).

El resultado de la acción es un documento único almacenado en el directorio asociado a los documentos del servidor (por ejemplo directorio /db para Alejandría).

Su caso de redireccionamiento funciona tomando del URL la ruta al documento para realizar una búsqueda en una tabla (archivo\_id) de la DB asociativa, esta retorna el identificador en el sistema de repositorio origen para construir un nuevo URL y redireccionarlo como un caso IR.

Ejemplos: URL directo al archivo

<http://www.saber.ula.ve/db/ssaber/Edocs/pubelectronicas/educere/vol4num11/articulo4-11-1.pdf>

- IR:

Se caracteriza porque su URL contiene un Query String iniciado con la variable que contiene el valor del identificador único de un registro del sistema de repositorio origen (Alejandría).

El resultado de la acción es un registro único perteneciente al sistema de repositorio origen (Alejandría).

Su caso de redireccionamiento funciona tomando del URL el identificador de registro del sistema de repositorio origen (Alejandría) para realizar una búsqueda en una tabla (asociacion\_id) de la DB asociativa, esta retorna el identificador en el sistema de repositorio destino (Handle, DSpace) para construir un nuevo URL y finalmente redireccionarlo.

Ejemplo: URL con acceso directo a un registro

[http://www.saber.ula.ve/cgi-win/be\\_alex.exe?Acceso=T016300000223/0&Nombrebd=ssaber](http://www.saber.ula.ve/cgi-win/be_alex.exe?Acceso=T016300000223/0&Nombrebd=ssaber)

- POST

Se caracteriza porque el método de envío es POST, es decir son formularios.

El resultado de la acción es cero, uno o más registros que contienen las palabras clave contenidas en las variables POST asociadas a la búsqueda.

Su caso de redireccionamiento funciona tomando el valor de las variables POST enviadas, para luego construir un nuevo Query String adaptado al sistema de repositorio destino (DSpace) y finalmente redireccionarlo.

Ejemplo: cajas de búsqueda de las unidades de investigación, revistas, postgrados, etc.

---

## 6. ANEXO 2: DESCRIPCIÓN DE LOS PASOS PARA LA OBTENCIÓN DE LOS DATOS DE ASOCIACIÓN (CASO ALEJANDRÍA-DSpace)

- Para generar el archivo doc\_acceso.sql:

Ingrese a la interfaz de Sybase y realice a la DB "SSABER" por separado cada una de las sentencias SQL contenidas en el archivo "sql\_alejandria.txt", el resultado serán dos archivos:

"archivotoacceso\_saber.txt" y "archivotocodigo\_saber.txt"; cópielos en el directorio "/tablas" de su conversor.

Corra el archivo "<raiz>/conversor/tablas/archivos.php" en un navegador para generar automáticamente el archivo "doc\_acceso.sql", listo para ser copiado a la tabla "archivo\_id" de la DB asociativa por medio del conversor

- Para generar el archivo acceso\_handle.sql:

Ingrese a la interfaz web de PostgreSQL y realice a la DB "DSpace" la sentencia SQL contenida en el archivo "sql\_dspace.txt" del directorio "/tablas" de su conversor, luego exporte el resultado con el comando "Bajar" de esa misma interfaz a un archivo de texto plano.

Al eliminar las comillas dobles (") de ese archivo y guardarlo con el nombre "acceso\_handle.sql" en el directorio "/tablas" de su conversor, el archivo estará listo para ser copiado a la tabla "asociacion\_id" de la DB asociativa por medio del conversor.

- Para generar el archivo comunidad\_handle.sql:

Ingrese a la interfaz web de PostgreSQL y realice a la DB "DSpace" la sentencia SQL contenida en el archivo "sql\_comunidades.txt" del directorio "/tablas" de su conversor, luego exporte el resultado con el comando "Bajar" de esa misma interfaz a un archivo de texto plano.

Al eliminar las comillas dobles (") de ese archivo y guardarlo con el nombre "comunidad\_handle.sql" en el directorio "/tablas" de su conversor, el archivo estará listo para ser copiado a la tabla "asociacion\_id" de la DB asociativa por medio del conversor.

---

Para cualquier información extra no dude en comunicarse con el creador del conversor al correo electrónico [tupac@ula.ve](mailto:tupac@ula.ve), [tupacamaru39@gmail.com](mailto:tupacamaru39@gmail.com)

## A.5 Cronograma de Trabajo

Se presenta la planificación de 6 meses (desde el 05/09/07 hasta el 29/02/08).

Nombre de tarea	Duración	Comienzo	Fin
<b>1 Primera Fase: Investigación</b>	<b>23 días</b>	<b>mié 05/09/07</b>	<b>vie 05/10/07</b>
1.1 Revisión de conceptos	8 días	mié 05/09/07	vie 14/09/07
1.2 Revisión de estadísticas	8 días	lun 17/09/07	mié 26/09/07
1.3 Revisión de antecedentes	7 días	jue 27/09/07	vie 05/10/07
<b>2 Segunda Fase: Análisis, Diseño e Implementación</b>	<b>70 días</b>	<b>lun 08/10/07</b>	<b>vie 11/01/08</b>
2.1 Análisis de los sistemas de consulta	10 días	lun 08/10/07	vie 19/10/07
2.2 Búsqueda de herramientas para redireccionar solicitudes Web	20 días	lun 22/10/07	vie 16/11/07
2.3 Aplicación de pruebas básicas de redireccionamiento	20 días	lun 05/11/07	vie 30/11/07
2.4 Diseño del diagrama general de la solución al problema	30 días	lun 19/11/07	vie 28/12/07
2.5 Implementación del convertor	30 días	lun 03/12/07	vie 11/01/08
<b>3 Tercera Fase: Instalación y Pruebas</b>	<b>15 días</b>	<b>lun 14/01/08</b>	<b>vie 01/02/08</b>
3.1 Instalación del convertor	2 días	lun 14/01/08	mar 15/01/08
3.2 Determinación de los casos de prueba	2 días	mar 15/01/08	mié 16/01/08
3.3 Pruebas del convertor en un escenario simulado	4 días	jue 17/01/08	mar 22/01/08
3.4 Documentación sobre la instalación y configuración del convertor	5 días	lun 21/01/08	vie 25/01/08
3.5 Presentación de la solución al equipo de migración del RI SABER-ULA	5 días	lun 28/01/08	vie 01/02/08
<b>4 Cuarta Fase: Documentación Final del Proyecto</b>	<b>128 días</b>	<b>mié 05/09/07</b>	<b>vie 29/02/08</b>
4.1 Documentación de antecedentes y marco teórico	30 días	mié 05/09/07	mar 16/10/07
4.2 Documentación de resultados y conclusiones	30 días	lun 21/01/08	vie 29/02/08

Planificación por fechas de las tareas del proyecto de grado

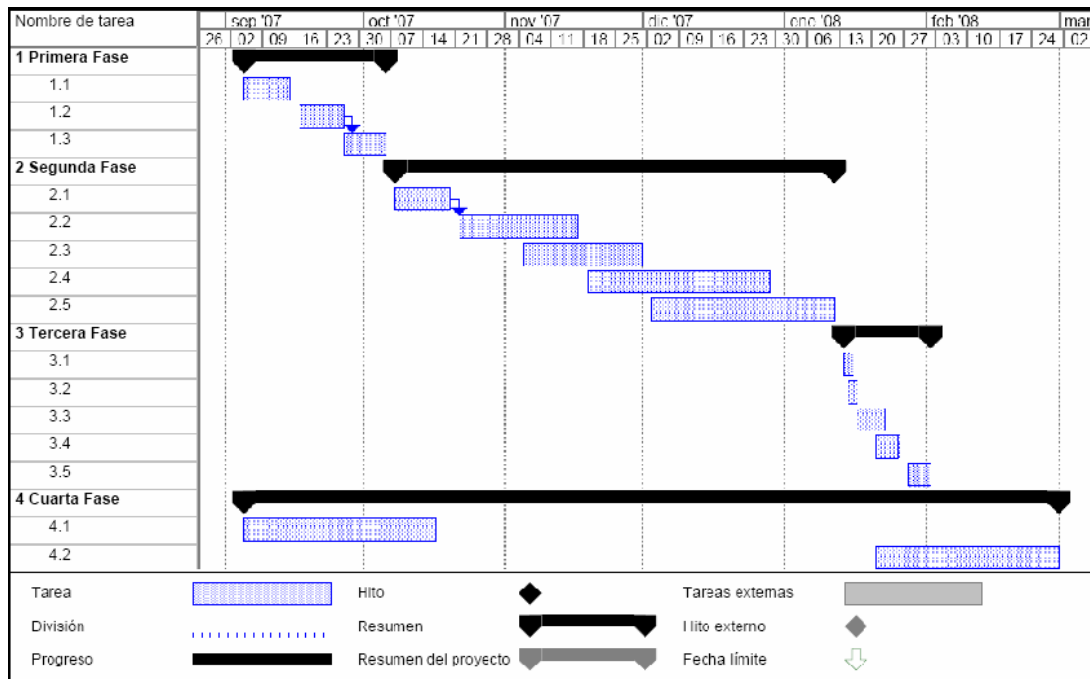


Diagrama de Gantt de las tareas del proyecto de grado

## Bibliografía y Referencias

- Ahmed, K., Ayers, D., et al. (2001). Professional XML Meta Data. Birmingham, UK: Wrox Press.
- Alejandría (2006) publicada bajo una licencia de Creative Commons. Especificaciones de la plataforma Alejandría. Consultado en oct., 11, 2007 en <http://alejandria.hacer.ula.ve/soporte/Especificaciones.htm>.
- Alejandría (2005). Manual de administración de Alejandría para repositorios institucionales. Consultado en abr., 17, 2008 en [http://www.saber.ula.ve/cgi-win/be\\_alex.exe?Nombrebd=SSABER&Opc=FD\\_EST;](http://www.saber.ula.ve/cgi-win/be_alex.exe?Nombrebd=SSABER&Opc=FD_EST;)
- Alejandría (2006). Plataforma Alejandría. Consultado en septiembre, 06, 2007 en <http://www.alejandria.biz/productos/PlataformaAlejandria/>.
- Apache.org (2008). ab - Apache HTTP server benchmarking tool. Consultado en mayo, 05, 2008 en <http://httpd.apache.org/docs/2.0/es/programs/ab.html>.
- Apache.org (1996). Apache mod\_rewrite. Consultado en nov., 17, 2007 en [http://httpd.apache.org/docs/2.0/mod/mod\\_rewrite.html](http://httpd.apache.org/docs/2.0/mod/mod_rewrite.html).
- Apache.org (1997). URL Rewriting Guide. Consultado en nov., 19, 2007 en <http://httpd.apache.org/docs/2.0/misc/rewriteguide.html>.
- Berners-Lee, Tim y Cailliau, Robert, (1990) "World Wide Web: Proposal for a HyperText Project". ECP. CERN.
- Berners, L. (1994). Uniform Resource Locators (URL). Consultado en oct, 30, 2007 en <ftp://ftp.isi.edu/in-notes/rfc1738.txt>.
- Bohen, R. (2006). Apress the definitive guide to apache mod\_rewrite. Nueva York, USA: apress.
- Consejo de Computación Académica CCA (2006). Plan institucional de liberación de software y migración a plataformas de software libre Universidad de Los Andes 2006-2010. Consultado en sep., 18, 2007 en [http://nux.ula.ve/documentos/Plan\\_de\\_Migracion\\_Mayo\\_2006\\_CCA.pdf](http://nux.ula.ve/documentos/Plan_de_Migracion_Mayo_2006_CCA.pdf).



- Dávila, J., Núñez, L., Sandía, B., Silva, J., Torrén, R. (2006) *www.saber.ula.ve: Un ejemplo de Repositorio Institucional Universitario*. *Interciencia* 31: 29-36. Consultado en: [www.saber.ula.ve](http://www.saber.ula.ve), [www.scielo.org.ve](http://www.scielo.org.ve), [www.interciencia.org/v31\\_01/](http://www.interciencia.org/v31_01/);
- DCMI (2006). *Dublin Core Metadata Element Set, version 1.1*. Consultado en octubre, 15, 2007 en <http://dublincore.org/documents/dces/>.
- DOI (2007). *The DOI System*. Consultado en oct., 12, 2007 en <http://www.doi.org/index.html>.
- DOI (2006). *DOIS*. Consultado en septiembre, 20, 2007 en <http://www.doi.org/>, <http://www3.interscience.wiley.com/doiinfo.html>, <http://www.handle.net/>.
- DSpace (2007). *Preguntas Frecuentes*. Consultado en noviembre, 13, 2007 en [http://www.dspace.org/index.php?option=com\\_content&task=view&id=65#how](http://www.dspace.org/index.php?option=com_content&task=view&id=65#how).
- Flanagan, D. (1997). *JavaScript: The Definitive Guide, Second Edition*. California, USA: O'Reilly.
- Fried Foster, N., Gibbons, S. (2005) *Understanding Faculty to Improve Content Recruitment for Institutional Repositories*. *D-Lib Magazine*. Volume 11 Number 1 ISSN 1082-9873
- *Gaceta Oficial de la República Bolivariana de Venezuela* (2004). Decreto 3390. Consultado en sep., 15, 2007 en <http://www.gobiernoenlinea.gob.ve/docMgr/sharedfiles/Decreto3390.pdf>.
- Handle.net (2005). *El Handle System*. Consultado en septiembre, 24, 2007 en <http://www.handle.net/>.
- Handle.net (2006). *HANDLE.NET (version 6.2): Technical Manual (version 1)*. : Corporation for National Research Initiatives.
- Interactive Programmers Community (2000). *DICCIONARIO INFORMÁTICO*. Consultado en abr., 19, 2008 en <http://www.lawebdelprogramador.com/diccionario/>.
- Laurie, B. y Laurie, P. (2002). *Apache: The Definitive Guide, Third Edition*. California, USA: O'Reilly.
- MEDRA (2004). *¿Qué es la international DOI Foundation?*. Consultado en octubre, 16, 2007 en <http://www.medra.org/en/faq.htm#faq6>.
- Mundogeek.net (2004). *JavaScript, el objeto Window*. Consultado en Oct., 11, 2007 en <http://mundogeek.net/archivos/2004/08/10/javascript-el-objeto-window/>.

- Nognu.org (2007). Glosario. Consultado en mayo, 01, 2008 en [http://www.nognu.org/emacs-man-es/index\\_577.html](http://www.nognu.org/emacs-man-es/index_577.html).
- Núñez, L. (2001). "Bibliotecas, autores, editores y estado". *Investigación*, 5, 46-51. Consultado en sep., 08, 2007 en <http://www.saber.ula.ve>
- Núñez, L.A. (2002) *La Reconquista Digital de la Biblioteca Pública*. *Interciencia*, 27, 195-200. Publicado en: <http://www.interciencia.org> y también en <http://www.saber.ula.ve>.
- PHP.net (2001). header. Consultado en may, 17, 2008 en <http://ve.php.net/header>.
- Spainhour, S., Eckstein, R. (1999) *Webmaster in a Nutshell*, Second Edition. O'Reilly.
- Sociedad Max Planck (2003). Declaración de Berlín sobre acceso abierto al conocimiento. Consultado en oct., 02, 2007 en <http://oa.mpg.de/openaccess-berlin/berlindeclaration.html>.
- Sommerville, I. (2002). *Ingeniería de Software*, Sexta Edición. Pearson Educación
- Soros (2002). Budapest Open Access Initiative. Consultado en oct., 06, 2007 en <http://www.soros.org/openaccess/read.shtml>.
- Sun, S., Reilly, S., Lannom, L. (2003) "Handle System Namespace and Service Definition". Internet Engineering Task Force (IETF) Request for Comments (RFC), RFC 3651.
- Tizag.com (2000). JavaScript Redirect. Consultado en Dic., 01, 2007 en <http://www.tizag.com/javascriptT/javascriptredirect.php>.
- Torrén R. (2003). *Desarrollo de Sistemas de Información bio-climática*. Tesis de Maestría del Postgrado en Computación de la Universidad de Los Andes.
- Universidad Carlos III de Madrid (2007). glosario. Consultado en mayo, 14, 2008 en <http://e-archivo.uc3m.es:8080/dspace/help/glosario.html>.
- Vera, C. (2003). *Sistema de Información Bibliográfica de la CEPAL: manual de referencia*. Santiago de Chile: Organización de las Naciones Unidas.
- W3C (1994). GET Method. Consultado en feb., 14, 2008 en <http://www.w3.org/Protocols/HTTP/Methods/Get.html>.
- W3C (1994). POST Method. Consultado en feb., 14, 2008 en <http://www.w3.org/Protocols/HTTP/Methods/Post.html>.
- Webdesign.com (2002). Meta Refresh Tag. Consultado en Marzo, 12, 2008 en <http://webdesign.about.com/od/metataglibraries/a/aa080300a.htm>.
- Webdesign.com (2001). Meta Tags. Consultado en Marzo, 11, 2008 en <http://webdesign.about.com/od/metatags/a/aa111600a.htm>.

- Webometrics (2008). World ranking on the Web: Top 200 Repositories. Consultado en mar., 03, 2008 en [http://www.webometrics.info/top200\\_rep.asp](http://www.webometrics.info/top200_rep.asp).
- Wyles, R. et al. (2006). Technical Evaluation of selected Open Source Repository Solutions. Nueva Zelanda: CPIT.