



Facultad 1

Versión 2.0 del proveedor de servicios OAI-PMH para el Sistema de Gestión de Contenidos Drupal

**Trabajo de diploma para optar por el título de
Ingeniero en Ciencias Informáticas**

Autor:

Ayane Suárez Romero

Tutores:

Ing. Maikel Manuel Fernández Fernández

Ing. Luis Domínguez Cruz

La Habana, Junio de 2012

“Año 54 de la Revolución”

DECLARACIÓN DE AUTORÍA

Declaro ser autor único del presente trabajo de diploma y concedo a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo. Para que así conste firmo la presente a los ____ días del mes de _____ del año_____.

Firma del Autor
Ayane Suárez Romero

Firma del Tutor
Ing. Maikel M Fernández Fernández

Firma del Tutor
Ing. Luis Dominguez Cruz



*“Por el grosor del polvo en los libros de una biblioteca pública, puede
medirse la cultura de un pueblo”*

John Ernest Steinbeck

AGRADECIMIENTOS

A mi tutor Maikel que se comportó como mi guía principal durante todo el proceso desarrollo de la tesis.

A la profesora Nayris que me esclareció diversos conceptos y de forma inconsciente el funcionamiento de mi sistema.

Agradecimientos especiales a mi madre Teresita que conoce mi trabajo al dedillo y me guió muchísimo en la investigación y redacción de mi tesis. A mi padre Gerardo que me apoyó en la búsqueda de información y a mi familia en general que funcionó como jurado y me ayudaron a corregir imprecisiones constantemente.

A Estrellita que con sus conocimientos en la rama de las Ciencias de la Información, compartió cuantiosa información junto a mí y hubo así una retroalimentación en este tema que gira alrededor de las bibliotecas digitales.

A todos mis compañeros de la universidad.

De forma general a todos aquellos que de una u otra forma apoyaron en mi formación.

DEDICATORIA

Dedico este trabajo de diploma de forma especial a toda mi familia. A mi madre y padre quienes han sido la base de mi formación, mi ejemplo a seguir y a los que tanto adoro. A mi hermano Emith que nos hemos apoyado el uno al otro durante muchos años en la convivencia. A Humberto que se ha comportado como un segundo padre. A mi abuela Tata, que la adoro muchísimo y que estoy muy contento y orgulloso de que haya logrado verme superar. Además se la dedico a quienes se convirtieron en mi segunda familia y de ella, en especial a Estrellita, una persona muy importante en mi vida.

RESUMEN

La presente investigación lleva como propósito describir la implementación de la segunda versión del módulo proveedor de servicios, desarrollado para el proyecto “Biblioteca Digital Alma Mater”; para la República Bolivariana de Venezuela. El módulo anterior estaba concebido para su funcionamiento sobre la arquitectura Drupal 6.x y tenía el propósito de garantizar el indexado de contenidos distribuidos bajo el protocolo Iniciativa de Archivos Abiertos - Protocolo para la Transmisión de Metadatos (OAI-PMH del inglés *Open Archives Initiative – Protocol for Metadata Harvesting*).

El presente desarrollo comparte la arquitectura y objetivos de su precedente pero incorpora nuevas funcionalidades y corrige otras ya existentes. El documento recoge un estado del arte que arroja importantes resultados sobre las tendencias actuales en el área de investigación y además expone el proceso de desarrollo siguiendo la metodología Proceso Unificado de Rational (RUP del inglés *Rational Unified Process*).

El protocolo OAI-PMH, la forma de describir los documentos, el cómo se establece la comunicación y las posibilidades de Drupal para incorporar soluciones de este tipo son aspectos que quedan recogidos en la investigación.

El sistema asegura el consumo de fuentes de contenidos de acceso abierto dando lugar a un crecimiento acelerado de los fondos en sistemas bibliotecarios, además de la mejora de los procesos de recuperación de información. El módulo desarrollado sigue los estándares de la arquitectura de Drupal 6.x e incorpora la gestión de repositorios, el indexado de colecciones y la actualización automática. De esta forma provee al usuario de un mecanismo mejorado para la indexación de contenidos.

Palabras clave: Acceso Abierto, *Dublin Core*, Drupal, OAI-PMH, Proveedor de servicios, Repositorio, Sistemas de información.

TABLA DE CONTENIDOS

INTRODUCCIÓN.....	1
Capítulo 1.....	6
ANÁLISIS CONCEPTUAL Y TECNOLÓGICO DE LAS BIBLIOTECAS Y EL PROTOCOLO OAI-PMH ..	6
1.1 Sistemas de Información.....	6
1.1.1 <i>Las bibliotecas como sistemas de Información</i>	7
1.2 El movimiento Open Access	8
1.3 Estándares de metadatos	10
1.3.1 <i>MARC</i>	13
1.3.2 <i>Dublin Core</i>	13
1.4 Protocolos para el intercambio de información en bibliotecas	14
1.4.1 <i>El protocolo Z39.50</i>	15
1.4.2 <i>El protocolo OAI-PMH</i>	15
1.4.2.1 Estructura de OAI-PMH	17
1.4.2.2 La comunicación OAI-PMH	19
1.4.2.3 Proveedor de datos.....	20
1.4.2.4 Proveedor de servicios	21
1.5 Sistemas para repositorios	22
1.6 Drupal para el desarrollo de sistemas de información tipo biblioteca.....	24
1.6.1 <i>Drupal</i>	24
1.7 Entorno tecnológico para el trabajo con Drupal	25
1.7.1 <i>Sistema de Gestión de Base de Datos</i>	25
1.7.1.1 MySQL	26
1.7.1.2 PostgreSQL.....	26
1.7.2 <i>Servidor Web Apache</i>	26
1.7.3 <i>Programación para la Web</i>	27
1.7.3.1 PHP	27
1.7.3.2 XML	27
1.7.3.3 RDF.....	28
1.7.3.4 XHTML	28

1.7.3.5	CSS.....	28
1.7.3.6	JavaScript	29
1.7.3.7	JQuery.....	29
1.7.3.8	Ajax	29
1.7.4	<i>Proceso de mejora de software</i>	29
1.7.4.1	Proceso.....	30
1.7.4.2	Mejora	30
1.7.4.3	CMMI.....	30
1.7.5	<i>Metodologías de desarrollo</i>	31
1.7.5.1	RUP.....	31
1.7.6	<i>Herramientas CASE</i>	32
1.7.6.1	NetBeans IDE	32
1.7.6.2	Visual Paradigm.....	32
1.8	Casos de estudio de bibliotecas desarrollados con Drupal	33
1.9	Propuesta de solución tecnológica.....	33
Capítulo 2.....		35
PROPUESTA DE SOLUCIÓN DEL PROVEEDOR DE SERVICIOS OAI PARA EL SISTEMA DE GESTIÓN DE CONTENIDOS DRUPAL.....		35
2.1	Objeto de automatización	35
2.2	Información que se maneja.....	35
2.3	Propuesta del sistema.....	35
2.4	Modelo de dominio	36
2.4.1	<i>Diagrama de clases del modelo del dominio</i>	37
2.4.1.2	Descripción de las clases del modelo del dominio	37
2.5	Especificación de requisitos de software.....	38
2.5.1	<i>Descripción de los actores</i>	38
2.5.2	<i>Requisitos funcionales</i>	39
2.5.3	<i>Requisitos no funcionales</i>	39
2.6	Descripción del sistema propuesto.....	43
2.6.1	<i>Listado de casos de uso del sistema</i>	43
2.6.2	<i>Diagrama de casos de uso del sistema</i>	46

2.6.3	<i>Patrones de casos de uso</i>	46
2.6.4	<i>Descripción de los casos de uso arquitectónicamente significativos del sistema</i>	47
Capítulo 3	51
CONSTRUCCIÓN DE LA PROPUESTA DE SOLUCIÓN Y VALIDACIÓN DE LOS RESULTADOS DEL PROVEEDOR DE SERVICIOS OAI PARA EL SISTEMA DE GESTIÓN DE CONTENIDOS DRUPAL . 51		
3.1	Análisis	51
3.1.1	<i>Diagrama de clases del análisis</i>	51
3.2	Diseño	52
3.2.1	<i>Arquitectura de software</i>	52
3.2.2	<i>Diagrama de clases</i>	53
3.2.3	<i>Diagrama de clases del diseño</i>	55
3.3	Diseño de la base de datos	57
3.3.1	<i>Diagrama de clases persistentes</i>	57
3.3.2	<i>Modelo de datos</i>	58
3.4	Modelo de implementación	59
3.4.1	<i>Diagrama de despliegue</i>	59
3.4.2	<i>Diagrama de componentes</i>	60
3.4.2.1	<i>Descripción de los componentes</i>	61
3.5	Diferencias entre las versiones de los módulos implementados	62
3.6	Pruebas	63
3.6.1	<i>Pruebas internas</i>	63
3.6.2	<i>Tiempo de recolección</i>	64
CONCLUSIONES		
RECOMENDACIONES.....		
REFERENCIAS BIBLIOGRÁFICAS		
BIBLIOGRAFÍA CONSULTADA		
GLOSARIO DE TÉRMINOS		
ANEXOS		
Anexo 1. Descripción de casos de uso del sistema.....		73
Anexo 2. Diagrama de clases del análisis		85
Anexo 3. Descripción de las tablas.....		87

Anexo 4. Diseño de casos de prueba basadas en casos de uso.....	90
Anexo 5. No conformidades.....	93
Anexo 6. Vistas del sistema.....	98

INTRODUCCIÓN

En el mundo actual la era de la información digital está en pleno auge y la generación de contenido científico y técnico está creciendo a un ritmo muy acelerado. Este gran cúmulo está disponible en diferentes formatos como: texto, audio o video, ejerciendo gran influencia en la ciencia, el conocimiento y la transferencia de tecnologías, con propósitos diversos como investigación, enseñanza y ocio, creados por personal autónomo, institucional, organizacional o corporativo, disponibles en todo momento del día.

Efectuar una búsqueda en *Internet* sobre un tema específico en ocasiones se convierte en un proceso tedioso debido a que los posibles resultados son muy diversos, de procedencia heterogénea, de variados autores e inclusive de contenidos redundantes. Esto dificulta en gran medida el hallazgo de información útil para un usuario determinado, generalmente con escasos conocimientos en técnicas de recuperación de información.

La proliferación de herramientas para buscar información en la Web, ha estado en constante crecimiento en los últimos años y ejemplo de ello se evidencia según lo citado por Stanley (1998), donde señala que en 1998 existían más de 2000 motores de búsqueda diferentes en la Web, mientras que en 1995 había tan solo una docena. En la actualidad existen numerosos motores de búsqueda tales como: Archie, Google, Altavista, Yahoo, Infoseek, Terra y Scirus. Cada uno con sus características, utilidades e interfaces particulares que las distinguen una de otras, trayendo como consecuencia que los usuarios en la mayoría de los casos, tengan que experimentar largas jornadas de búsqueda en múltiples entornos, familiarizándose con cada uno de ellos por separado, retrasando y complicando el proceso de búsqueda.

Con el objetivo de facilitar el acceso a la información y ponerla al servicio de otros surge el manejo de datos e informaciones, siendo esto, uno de los aspectos más importantes para cualquier entidad. Este manejo, engloba diferentes actividades como recolección, almacenamiento, recuperación y propagación hacia distintos lugares y personas, así como su uso para varias actividades dentro de una organización.

Las tecnologías empleadas para la organización, representación y distribución de la documentación constituyen una gama muy amplia y entre ellas se señalan: aplicaciones para el manejo de repositorios, estándares para el intercambio de metadatos, formatos de metadatos para describir los documentos, lenguajes de programación y sistemas de gestión de contenidos (CMS del inglés *Content*

Management Systems), entre otros. En la actualidad los disímiles tipos de sistemas de información (SI) no sólo se encargan de implementar el ciclo básico de un documento, sino que deben garantizar su distribución de forma que puedan ser consumidos por todos, independientes de la tecnología. Debido a los aspectos mencionados con antelación, la estandarización de la pluralidad tecnológica representa un papel esencial en estos procesos.

Los contenidos científicos también se ven afectados por toda la problemática anterior, por ello se han tomado una serie de acciones para facilitar a los usuarios la búsqueda de información, pero siguiendo principios propios de estos tipos de recursos.

Para la distribución de contenidos de corte científico e investigativo a través de las redes informáticas, se han desarrollado protocolos como: Aplicación de Servicios de Definición y Especificación de Protocolo (Z39.50 del inglés *Application Service Definition and Protocol Specification*) y la Iniciativa de Archivos Abiertos - Protocolo para la Transmisión de Metadatos (OAI-PMH del inglés *Open Archives Initiative – Protocol for Metadata Harvesting*). También esquemas de metadatos para la descripción de recursos entre los que se hallan: *Dublin Core Metadata Element Set*, *Visual Resources Association Core Categories* (VRACore), *Encoded Archival Description* (EAD) y MARC.

Uno de los protocolos más popularizado es el OAI-PMH, basado fundamentalmente en la filosofía del libre acceso (*Open Access*) a la información. La cantidad de repositorios y revistas científicas digitales que presentan sus contenidos bajo OAI-PMH es muy amplia y con tendencia al crecimiento. Dentro del mismo participan dos roles, el proveedor de datos (DP del inglés *data provider*) y el proveedor de servicios (SP del inglés *service provider*). El desarrollo de un proveedor de servicios permite contar con una herramienta para consumir la información, recopilando los metadatos de variados archivos provenientes de numerosos proveedores. Esto garantiza que la búsqueda de información sea gestionada en la propia base de datos donde se realice la indexación de información, garantizando acceso amplio a la misma.

La tendencia actual de las bibliotecas digitales es la coexistencia, donde cada una comparte su contenido para que otras puedan servirse del mismo, proporcionando mayor dinamismo y facilidad en generación y recuperación de información. Existen herramientas libres para la gestión de procesos bibliotecarios, como Koha, Openbiblio, Gnuteca y PHPmyBibli, entre otros. También entre las tecnologías utilizadas para estos propósitos se encuentran los CMS, que permiten administrar de forma dinámica recursos Web, uno de los más divulgados es Drupal.

Para el desarrollo de un entorno bibliotecario, la versión seis del CMS Drupal presenta una serie de módulos, entre ellos un proveedor de datos OAI. Para el funcionamiento de un SP existe una versión de un módulo llamado “datapviders” desarrollado para el proyecto de “Biblioteca Digital Alma Mater”, destinado al Ministerio del Poder Popular para la Educación Universitaria (MPPEU) en la República Bolivariana de Venezuela.

Este módulo implementa al protocolo OAI-PMH, pero en su funcionamiento se desaprovechan ciertas ventajas que el protocolo y Drupal brindan como: el proceso de indexado es manual, el módulo depende de que previamente exista una estructura para la gestión de documentos, no cuenta con la indexación basada en colecciones, no se analizan los estándares de metadatos de la fuente, bajo control de contenidos duplicados y de validaciones de datos y uso escaso de los verbos que proporciona el SP.

A partir de las dificultades presentes en la implementación del módulo “datapviders”, se origina el **problema a resolver**, formulado en la siguiente interrogante: ¿Cómo perfeccionar el funcionamiento del módulo del proveedor de servicios OAI-PMH desarrollado para la versión seis del CMS Drupal?

El **objeto de estudio** se orienta hacia los SI y los protocolos para el intercambio, transferencia y recuperación de información bibliográfica, enmarcando el **campo de acción** en el protocolo OAI-PMH.

El **objetivo general** del presente estudio, es implementar la versión 2.0 del módulo proveedor de servicios OAI-PMH, para la versión 6.x del CMS Drupal, para garantizar el intercambio de información con fuentes de acceso abierto.

Los **objetivos específicos** que se proponen en la investigación son:

- Realizar un estado del arte sobre los protocolos y normas para el intercambio, transferencia y recuperación de información científica y los SI tipo biblioteca.
- Reflejar las tecnologías empleadas a nivel mundial en el empleo de dicho protocolo en la gestión de bibliotecas digitales.
- Describir el funcionamiento del protocolo y representar sus flujos de trabajo.
- Diseñar nuevas estructuras y funcionalidades para el módulo OAI versión 1.0 e implementar las nuevas funcionalidades, identificadas para la versión 2.0 del módulo OAI.
- Comprobar el funcionamiento mejorado de la solución.

Se **defiende la idea** de que desarrollando un módulo para un portal de biblioteca que implemente el protocolo OAI-PMH, se puede aumentar de forma acelerada sus fondos, mediante procesos de indexación y comunicación con otras fuentes de acceso abierto, lo que brindaría una gran ventaja en la recuperación de información a los usuarios.

Para dar cumplimiento al objetivo general se planificaron las siguientes **tareas de investigación**:

- Indagar sobre los protocolos y normas para la distribución de información en SI.
- Investigar sobre el CMS Drupal y sus experiencias en el desarrollo de proyectos tipo bibliotecas o archivos.
- Describir el funcionamiento del protocolo OAI-PMH.
- Identificar y describir las funcionalidades de un proveedor de servicios OAI.
- Analizar deficiencias presentes en el módulo creado anteriormente.
- Mejorar el desarrollo anterior mediante recolección de documentos por colecciones, tareas automáticas, validación de datos y perfeccionamiento de interfaz visual.
- Integrar el desarrollo con la arquitectura de Drupal.
- Realizar test de funcionamiento del componente desarrollado.
- Comparar el tiempo de demora de indexado de documentos entre ambas versiones del módulo.

Los **métodos científicos de investigación** empleados son:

Análítico-sintético: método teórico cuya finalidad accedió al estudio de las características principales de los proveedores de servicios, del estándar de metadatos DC y de fuentes de libre acceso.

Análisis histórico-lógico: método teórico que permite realizar un estudio profundo de la bibliografía existente acerca del desarrollo y evolución de los SI, los SID, el uso de OAI-PMH en estos sistemas y las tecnologías que garantizan el correcto funcionamiento del SP.

Entrevista: Para esto se sostuvieron conversaciones con personal con previos conocimientos sobre el tema con el fin de comprender el funcionamiento del protocolo, de los proveedores y del funcionamiento del módulo anteriormente desarrollado.

Observación: método empírico cuya función primaria es recoger información sobre el objeto que se toma en consideración. El mismo fue empleado en la detección de deficiencias que presenta el SP implementado con anterioridad y establecer nuevas funcionalidades a mejorar.

Los **resultados esperados** son:

- Aumento de las opciones para indexación de información bajo el protocolo OAI-PMH para el CMS Drupal.
- Reducción del tiempo del proceso de indexación de proveedores de datos OAI.
- Aumento de las fuentes de datos que acepta el sistema.
- Indexación automática de información mediante tareas cron.
- Recuperación de información de interés para los usuarios de forma rápida.

El contenido del documento está dividido en tres capítulos:

Capítulo I. Fundamentación teórica: Está orientado al fundamento teórico de la investigación, conceptos, definiciones, tecnologías y casos de estudios relacionados con el sistema propuesto. Ejemplo de ello, se cita a Drupal como CMS capacitado para ejercer el funcionamiento de un proveedor de servicios OAI.

Capítulo II. Propuesta de solución: Detalla la propuesta de solución del sistema, describiendo las reglas del negocio asociados al objeto de estudio y de automatización. Se da a conocer la propuesta del sistema y los diagramas para apoyar la comprensión del funcionamiento del mismo.

Capítulo III. Construcción de la propuesta de solución y validación de los resultados: Desarrollo y construcción de la propuesta de solución. Se muestran los diseños de la solución con ayuda de los diagramas de clases, de clases del diseño y modelo de datos. Se diseñan, describen y realizan los casos de pruebas aplicados al sistema para evaluar su desempeño funcional. Se establecen comparaciones entre las versiones de los módulos.

Capítulo 1

ANÁLISIS CONCEPTUAL Y TECNOLÓGICO DE LAS BIBLIOTECAS Y EL PROTOCOLO OAI-PMH

En el presente capítulo se efectúa un estudio de los aspectos más importantes relacionados con las bibliotecas como Sistemas de Información (SI), los estándares de metadatos, los protocolos para el intercambio de información, orientado en la Iniciativa de Archivos Abiertos - Protocolo para la Transmisión de Metadatos (OAI-PMH del inglés *Open Archives Initiative – Protocol for Metadata Harvesting*). Se aborda además acerca del uso de tecnologías, metodologías y herramientas de apoyo que intervienen en el proceso de construcción del producto final.

1.1 Sistemas de Información

Los sistemas que trabajan con elementos informativos, dígase datos, documentos, objetos, bibliografías, entre otros, se denominan SI. Según Muñoz (1999), en su reporte plantea que “un sistema de información es un conjunto de elementos o componentes relacionados con la información que interaccionan entre sí para lograr un objetivo: facilitar y/o recuperar información”.

Los Sistemas de Información Documental (SID) corresponden a un tipo especial de SI cuyo propósito principal es generar información útil a una necesidad específica de información suministrada por un usuario, con el fin de producir nuevo conocimiento. La información generada por un SID se representa mediante documentos que son manejados por el sistema como objetos de información digital (por ejemplo, en los Sistemas Integrados de Gestión de Bibliotecas (SIGB) y en las actuales bibliotecas digitales) (Guajardo, 2010).

A partir de la evolución de las tecnologías de información y comunicación y del auge de los SI, surge la necesidad de establecer entre los SIDs la interoperabilidad. Esto se hace con el fin de posibilitar la transferencia e intercambio de información entre sistemas.

La interoperabilidad es “la capacidad de un SI de comunicarse y compartir datos, información, documentos y objetos digitales de forma efectiva (con una mínima o nula pérdida de su valor y funcionalidad), con uno o varios SI (siendo generalmente estos sistemas completamente

heterogéneos, distribuidos y geográficamente distantes), mediante una interconexión libre, automática y transparente, sin dejar de utilizar en ningún momento la interfaz del sistema propio” (Gómez, 2007).

Estos sistemas llevan como principios básicos: facilitar el acceso a la información, alentar el aprendizaje y la curiosidad, contribuir a eliminar la memorización y estar al servicio de otros. Dichos sistemas se caracterizan por mantener su condición de sistema abierto, su capacidad para responder a los cambios, adaptarse a su ambiente, mantener cierta estabilidad y por su complejidad.

Los SI son considerados un componente vital para el desarrollo de cualquier entidad, máxime cuando se trata de una entidad con fines académicos y de investigación como son las universidades. Cualquier institución universitaria debe asumir que la información documental es la principal fuente de conocimientos para estudiantes, profesores e investigadores y que las bibliotecas, los centros de documentación y otras unidades de servicios de información son los entes que soportan toda la actividad intelectual (Calderín, 2007).

La gran mayoría de las actividades que se desarrollan hoy en día dependen de la información, constituyendo elementos de la vida diaria de cualquier país, donde un gran número de instituciones, organizaciones, agencias, instituciones académicas, líneas hoteleras y aéreas, entre otras, dependen de grandes SI que respaldan sus operaciones. Entre los más difundidos, se encuentran las bibliotecas, museos, centros de documentación e información, aulas especializadas, repositorios, revistas, sistemas de gestión documental y archivos y SI para la dirección (Ponjuan, y col., 2004).

1.1.1 *Las bibliotecas como sistemas de Información*

La palabra biblioteca deriva del latín *bibliothēca* ésta a su vez lo hace del vocablo griego *biblíon* (libro).

Según la norma UNE 50113-1:1992 sobre conceptos básicos de información y documentación, el término biblioteca puede definirse en dos sentidos:

- Cualquier colección organizada de libros y publicaciones en serie impresos u otros tipos de documentos gráficos o audiovisuales disponibles para el préstamo o consulta.
- Organismo, o parte de él, cuya principal función consiste en construir bibliotecas, mantenerlas, actualizarlas y facilitar el uso de los documentos que precisen los usuarios para satisfacer sus

necesidades de información, investigación, educativas o de esparcimiento, contando para ello con un personal especializado.

La norma UNE-EN ISO 2789:1996 sobre estadísticas internacionales de bibliotecas, la define como: organización o parte de ella cuya principal función consiste en mantener una colección y facilitar, mediante los servicios del personal, el uso de los documentos necesarios para satisfacer las necesidades de información, de investigación, de educación y ocio de sus lectores.

Las grandes bibliotecas cuentan también con departamentos especiales como hemerotecas (Según el diccionario de la Real Academia Española (RAE) una hemeroteca es una biblioteca en que principalmente se guardan y sirven al público diarios y otras publicaciones periódicas), además, poseen salas o dependencias en las que el público puede tener acceso a fotografías, videos, discos musicales, discos de computación, etc. y hasta ofrecen información sobre sus colecciones a través de *Internet*.

Las mismas, componen la principal fuente de documentación para la generación de nuevo conocimiento, por lo que a través de los años se han convertido en los centros de preservación documental más importantes. Por lo tanto, el uso e implementación de la tecnología es esencial para que el conocimiento sea mundialmente difundido.

Dentro de la clasificación de los tipos de bibliotecas, existen diversas tipificaciones, donde la mayoría apuntan fundamentalmente a cinco prototipos: nacional, pública, académica, especializada y escolar. Existen también otros criterios de clasificación, entre los que se pueden citar las relacionadas con el nivel de integración con las nuevas tecnologías de información y comunicación, entre otras. Estas son las bibliotecas: tradicionales, electrónicas, digitales, híbridas y virtuales.

Todos los tipos de bibliotecas, de una manera u otra, constituyen elementos fundamentales en el desarrollo cultural y científico de la sociedad.

1.2 El movimiento Open Access

La transmisión de ficheros por medio de *Internet* y las publicaciones electrónicas garantizan que el intercambio de documentación científica y su acceso sean cada vez más rápidos y hacederos. La aparición y crecimiento simultáneo de la gran red de redes, ha ocasionado un cambio revolucionario en las publicaciones y en general en la difusión del conocimiento. Sin embargo, a pesar de que las facilidades tecnológicas han aumentado inmensurablemente, las posibilidades que ofrecen las nuevas

tecnologías para las publicaciones científicas están siendo restringidas por barreras, en su mayoría económicas, que interponen las grandes empresas editoriales, encargadas de publicar la mayor parte de revistas científicas.

Se producen circunstancias como:

- Los científicos producen y ofrecen gratuitamente, los artículos que enriquecen a las revistas, para que posteriormente se les deniegue el acceso libre a sus propios contenidos.
- El abusivo aumento de los precios de las suscripciones a publicaciones periódicas y a bases de datos.
- La imposibilidad de crear bases de datos sin violar leyes de *copyright* que acoplen información de distinta procedencia para ser utilizadas en aplicaciones de búsqueda.

“Con respecto a los precios y de acuerdo con las estadísticas de la Asociación de Bibliotecas de Investigación (ARL del inglés *Association of Research Libraries*), el incremento de precios por suscripción a publicaciones periódicas en el período 1986-2002 siguió una curva galopante”. Esta subida por encima de los presupuestos de las bibliotecas, ha ocasionado que se produzca una disminución en los títulos adquiridos y un descenso en el presupuesto dedicado a otros servicios, incrementándose aún más, la brecha existente en el acceso a la información (Melero, 2005).

Todas estas circunstancias han provocado una reacción en cadena de científicos e instituciones denunciando la falta de accesibilidad a sus propias publicaciones debido a las barreras económicas impuestas por los monopolios editoriales, desarrollándose numerosas manifestaciones e iniciativas en contra de dichas restricciones, consolidadas en el llamado movimiento *Open Access* (OA), por sus siglas en inglés.

Con respecto a las publicaciones científicas, el OA está vinculado a las iniciativas o proyectos que favorezcan y promuevan el acceso abierto y sin restricciones a los trabajos publicados por la comunidad científica.

Existen dos definiciones de OA consideradas como referente de: la Declaración de Budapest (*Budapest Open Access Initiative*) en 2002 y la Declaración de Bethesda (*Bethesda Statement on Open Access Publishing*) en 2003. La primera define OA como "disponibilidad gratuita en la Internet pública, para que cualquier usuario la pueda leer, descargar, copiar, distribuir, imprimir, con la posibilidad de

buscar o enlazar todos los textos de estos artículos, recorrerlos para indexación exhaustiva, usarlos como datos para software, o utilizarlos para cualquier otro propósito legal, sin barreras financieras, legales o técnicas, respetando las leyes de *copyright* existentes”. “La segunda definición asume lo anterior complementándola con la garantía de que el copyright no será la barrera para el acceso a los artículos y que los ficheros de los mismos se depositarán en un repositorio institucional” (Melero, 2005).

Para apoyar a este movimiento, han surgido numerosas iniciativas y declaraciones internacionales con la premisa de facilitar el acceso a las publicaciones científicas por medios electrónicos, entre las que se citan: *BioMed Central* en 2001, *Budapest Open Access Initiative* en 2002, *The Public Library of Science* en 2000), *The Scholarly Publishing and Academic Resources Coalition* en 1998, *Scientific Electronic Library Online* en 2002, *Directory of Open Access Journals* en 2003 y la Declaración de Berlín (*Berlin Declaration on Open Access to Knowledge in the Sciences and Humanities*) en 2003.

Todas estas iniciativas internacionales, han surgido con un único objetivo común: mejorar el sistema tradicional de comunicación científica y facilitar el libre acceso a las publicaciones científicas a través de *Internet*. Esta eliminación de barreras favorece la visibilidad y la difusión de la investigación, enriquece la educación, rompe las barreras entre países pobres y ricos y hace que se recupere parte de la financiación con fondos públicos dedicados a la investigación científica.

Para que este movimiento OA pueda funcionar, se necesita de normas que tecnológicamente definan su funcionamiento y aquí destacan los estándares de metadatos y los protocolos de comunicación.

1.3 Estándares de metadatos

Un estándar establece un sistema común de terminología y definiciones para documentar datos, normando la forma de describirlos. Idealmente las estructuras y definiciones de metadatos deben tener su referencia en un estándar.

- Los estándares permiten la localización rápida de ciertos elementos. Si se utiliza un estándar, encontrar la información específica en un catálogo de metadatos será mucho más fácil que si no se utiliza ninguno.
- Posibilitan búsquedas automatizadas.

- Ayudan a minimizar la duplicación de esfuerzos en la elaboración, recolección, procesamiento o distribución de la información (Méndez y Torrén, 2003).

Jack Myers, de “*The Metadata Company*” afirma haber acuñado el término metadato en 1969 a pesar de que aparece en el libro, “*Extension of programming language concepts*”, publicado en 1968, por Philip R. Bagley (Howe, 2010). Una descripción muy usada sobre este término es con la definición de “datos sobre los datos”, que en términos generales no deja muy claro su significado. Un metadato son datos usados para describir otros datos.

En el ámbito de las ciencias de la información, se puede definir a los metadatos como información estructurada, sobre un recurso de información soportado en cualquier medio o formato. La información estructurada puede o no ser electrónica y los recursos descritos pueden ser publicaciones impresas, electrónicas, objetos digitales, etcétera (BNC, 2009).

Existen dos características que permiten identificar y definir los datos como metadatos. Una, es que la información debe estar estructurada, o sea, los datos deben seguir un esquema de metadatos reconocido y documentado. Otra característica es que los metadatos deben describir un recurso de información.

Entre sus funciones primordiales se encuentran: proveer un medio para descubrir que datos existen y cómo podrían ser obtenidos o accedidos y suministrar un mecanismo de búsqueda para coleccionar metadatos. Documentar contenido, calidad y rasgos o características de un conjunto de datos y así dar una indicación de la propiedad, idoneidad o correspondencia de uso (Gálvez, 2006).

Se entiende por metadatos, según (Méndez y Senso, 2004) como toda aquella información descriptiva sobre el contexto, calidad, condición o características de un recurso, dato u objeto que tiene la finalidad de facilitar su recuperación, autenticación, evaluación, preservación y/o interoperabilidad.

Los metadatos según (Lazinger, 2001) se han tipificado en general como: Descriptivos, Administrativos y Estructurales:

Los metadatos descriptivos, sirven para propósitos de recuperación (cómo se busca y ubica un recurso), de identificación (cómo se puede distinguir un recurso de otro similar) y selección (cómo determinar que un recurso responda a una necesidad en particular, por ejemplo, la versión Disco Versátil Digital (DVD del inglés *Digital Versatile Disc*) de un registro de video). También se pueden usar

los metadatos descriptivos para indicar la colocación (tener juntas todas las versiones de una obra) y la adquisición.

Los metadatos administrativos contienen información que pretende facilitar el manejo de los recursos. Pueden incluir información referida a cuándo y cómo fue creado un objeto, quién es el responsable de controlar su acceso o archivar su contenido, qué actividades de procesamiento y control se realizan y qué restricciones de uso o acceso se aplican a él.

Los metadatos estructurales pueden ser considerados como el aglutinante que mantiene juntos los componentes de los objetos digitales. Un libro por ejemplo, puede tener varios capítulos, cada uno de ellos consiste de un conjunto de páginas, cada página representada por un archivo digital separado. Se requiere entonces, metadatos estructurales para registrar las relaciones entre los archivos físicos y las páginas, entre páginas y capítulos y entre capítulos y el libro como un todo. También pueden documentar el orden y formato de elementos de datos en un conjunto de datos numéricos o estadísticos, como en un censo.

Los esquemas de metadatos son conjuntos de elementos de metadatos y reglas para su uso, definidos para un propósito en particular. Ejemplos de ellos tomados de BNC (2009) son:

- *Dublin Core Metadata Element Set*: es un conjunto de quince elementos de datos de descripción, con la intención de ser aplicables a cualquier tipo de recurso. Actualmente se ha constituido en una norma NISO: ANSI/NISO Standard Z39.85.
- *Visual Resources Association Core Categories (VRA Core)*: desarrollado principalmente para describir ítems de colecciones de recursos visuales (fotografías, dispositivos y/o imágenes digitales) de obras originales sobre arte y arquitectura.
- *Encoded Archival Description (EAD)*: desarrollado como un medio de representar ayudas para la recuperación archivística en formato electrónico.
- *MARC*: empleado para la descripción de datos bibliográficos de material impreso y manuscrito, archivos de computadora, mapas, música, seriadas, material visual y mezclas de materiales. La información bibliográfica incluye título, nombres, temas, notas, datos de publicación e información sobre descripción física de los artículos.

1.3.1 MARC

MARC (*Machine Readable Cataloging*): Es un estándar de amplio uso para la representación e intercambio de datos, tanto bibliográficos como de autoridad bibliográfica, de existencias, de clasificación y de información a la comunidad en una forma legible por la máquina. Desarrollado por la *Library of Congress*, ha sido una de las herramientas de mayor uso en el ámbito de las bibliotecas, sobre todo físicas y los servicios de información, tanto para crear como para intercambiar registros creados en sistemas de automatización de recursos de información.

1.3.2 Dublin Core

Dublin Core (DC) es uno de los esquemas más conocidos y usados actualmente en los servicios de información y el más usado en el movimiento OA. Es un conjunto eficaz de elementos para describir una extensa gama de recursos de red.

Este estándar se creó a partir de un taller de metadatos realizado precisamente en la ciudad de Dublin, Ohio, Estados Unidos (EU o EEUU) en 1995. La primera versión generó grandes expectativas, pues se componía únicamente por un pequeño conjunto de descriptores con los cuales se podía describir en parte y de forma muy sencilla un recurso. Ya en el año 2001 se aprobó como norma estatal el conjunto de elementos de *Dublin Core* según el organismo de normalización de los EU, dando lugar a la norma Z39-85:2001 DUBLIN CORE METADATA SET ELEMENT. Con el auge de *Internet* y principalmente por su nivel de penetración en el mercado y en el ambiente académico, se comenzó a hacer un uso cada vez más extenso del estándar, razón por la cual en el 2003 se aprobó como norma ISO 15836, basándose ciertamente en la norma Z39-85. Desde la creación de este estándar surgió el DCMII (*Dublin Core Metadata Initiative*), organización cuya misión es proveer estándares para buscar, compartir y administrar información, basándose en los principios como la construcción bajo consensos abiertos, neutralidad tecnológica y modelos de negocios (Siza, 2010).

La norma DC conlleva dos niveles: Simple y Cualificado. EL DC Simple presenta quince elementos agrupados en tres grandes conjuntos (contenido, propiedad intelectual e instanciación) como se muestran en la siguiente tabla:

Tabla 1.1. Elementos de Dublin Core Simple

Contenido	Propiedad Intelectual	Instanciación
Title	Creator	Date
Subject	Publisher	Type
Description	Contributor	Format
Source	Rights	Identifier
Language		
Relation		
Coverage		

El DC cualificado en general puede utilizar todas las posibilidades del modelo abstracto. En particular presenta adicionalmente los elementos audiencia [Audience], Procedencia [Provenance], Titulares de los derechos [RightsHolder], así como un grupo de elementos de matización (denominados por ello, calificadores) que refinan la semántica de los elementos de tal forma que pueden ser útiles para la recuperación/localización de recursos en *Internet* [resourcediscovery]. La semántica del DC ha sido establecida por un grupo internacional e interdisciplinario de profesionales de la biblioteconomía, la informática, la codificación textual, la comunidad museística y otros campos teórico-prácticos relacionados (BNC, 2009).

1.4 Protocolos para el intercambio de información en bibliotecas

En los SID, la capacidad de interoperar entre ellos está dada por el uso de protocolos de intercambio, transferencia y recuperación de información bibliográfica. Son numerosos los protocolos desarrollados a través de los años con el fin de permitir la interoperabilidad entre SID, entre ellos:

- Application Service Definition and Protocol Specification (Z39.50).
- Simple Digital Library Interoperability Protocol (SDLIP).
- OAI-PMH.
- Guildford protocol.
- Dienst protocol.
- ZING SRU/W.
- Starts (*Stanford Protocol Proposal for Internet Retrieval and Search*).
- Lyceum Protocol.

- Harvest: A Distributed Search System.

Sin embargo, sólo algunos de éstos se han consolidado internacionalmente como estándares aceptados por los profesionales de la información, son usados de manera masiva y su desarrollo y mantención se encuentra avalado por iniciativas tanto públicas como privadas: el Protocolo Z39.50 y el Protocolo OAI-PMH. Este último se encuentra acreditado por la Convención de Santa Fe como solución para garantizar la interoperabilidad sobre información de acceso pleno, por la Iniciativa de Archivos Abiertos (OAI del inglés *Open Archives Initiative*) que extendió su uso para la comunicación e intercambio de metadatos de materiales digitales, la *Digital Library Federation* y la *Coalition of Networked Information* en EU que ofrecen soporte organizacional a OAI.

1.4.1 El protocolo Z39.50

El protocolo Z39.50 fue desarrollado por la NISO (*National Information Standards Organization*), para normalizar la comunicación entre dos computadores y recuperar información entre sí. Basado en la arquitectura cliente/servidor, especifica los procedimientos, formatos y funciones necesarias para recuperar la información en bases de datos (ANSI/NISO, 2003).

El protocolo Z39.50 se generó en EU a partir de 1970 con la idea de desarrollar una aplicación para hacer la catalogación de forma compartida, de manera de estandarizar e implementar las funciones de recuperación de información en un ambiente de sistemas distribuidos. Este programa fue llamado el “*Linked Systems Project*” y agrupó a la Biblioteca del Congreso de EEUU, el Centro de Bibliotecas Computarizadas en Línea (OCLC) y la Red de Información para Investigación en Bibliotecas (Guajardo, 2010).

Durante los 80 y los 90, este protocolo fue de gran éxito, debido a la creación de catálogos bibliográficos, catalogación y gestión de los documentos residentes en el acervo bibliográfico de las instituciones, donde el tratamiento manual mediante fichas bibliográficas era un proceso complejo, sistematizando a su vez el uso de metadatos en formato MARC (Siza, 2010).

1.4.2 El protocolo OAI-PMH

El protocolo OAI-PMH es una sencilla interfaz que hace posible el acceso a metadatos de contenidos de distintas fuentes (Gómez, 2005), genera y promueve estándares que facilitan la difusión, el intercambio y la accesibilidad a documentos (De Giusty, 2006), apoyándose principalmente en la

creación de repositorios. Es una herramienta de interoperabilidad que posibilita el intercambio de metadatos sobre cualquier material almacenado en soporte electrónico.

OAI-PMH es en esencia una plataforma simple para acceder a la información bibliográfica disponible en un archivo o repositorio. De esta forma, cualquier institución puede realizar una implementación del protocolo para poner a disposición su propia base de datos o catálogo y a la vez consultar otras.

Respecto a su nombre, se debe indicar que el término archivo, refleja los orígenes de la iniciativa en el seno de las comunidades de publicaciones electrónicas, donde es sinónimo de depósito de documentos científicos a texto completo, utilizándose con el sentido amplio de depósito para almacenar cualquier tipo de información; no dice relación con el concepto tradicional de archivo con connotaciones de preservación y conservación. El término abierto se refiere al punto de vista de la arquitectura del sistema, en tanto se trata de definir interfaces que faciliten la disponibilidad de contenidos procedentes de una variedad de proveedores.

OAI se creó con la misión de desarrollar y promover estándares de interoperabilidad, para facilitar la difusión eficiente de contenidos en *Internet*. Surgió como un esfuerzo para mejorar el acceso a archivos de *eprints* (documentos científicos, tesis doctorales, artículos ya publicados, artículos pendientes de publicación, u otros materiales que pueden ser depositados libremente por los autores en repositorios), incrementando la disponibilidad de las publicaciones científicas. La OAI no es solamente un proyecto centrado en publicaciones científicas, sino en la comunicación de metadatos sobre cualquier material almacenado en soporte electrónico. No hay nada en el protocolo que impida a los implementadores, transmitir el contenido propiamente dicho de esos materiales (BNC, 2009). La reunión de Santa Fe, Nuevo México, USA en 1999, marca el punto de partida de la creación del OAI y el resultado de ella fue un conjunto de acuerdos técnicos y organizativos conocidos como la Convención de Santa Fe que estableció la recolección de metadatos como solución para la generación de interoperabilidad entre los distintos proveedores de *eprints*. Los aspectos técnicos de la convención incluían tres puntos fundamentales: un formato para los metadatos, un protocolo de intercambio y un sistema de identificación (Barrueco, 2006).

En el año 2000, se evidenció el interés en la iniciativa más allá de las comunidades científicas, ya que bibliotecarios y museólogos se mostraron interesados en descubrir formas de hacer visibles a los motores de búsqueda en *Internet* parte de las colecciones de bibliotecas y museos. Fruto de este interés se decidió ampliar el objeto de trabajo más allá de los *eprints* para incluir disciplinas que no

tuvieran este tipo de documentación. La *Digital Library Federation* y la *Coalition of Networked Information* en EU ofrecieron soporte organizacional a la iniciativa y se establecieron dos comités, uno de gestión y otro técnico que se encargaron de la coordinación del proyecto. Las especificaciones técnicas modificadas fueron hechas públicas en 2001 con la publicación del protocolo OAI-PMH versión 1.0, momento en que se inició la implementación del protocolo y aparecieron las primeras instituciones que lo utilizaron para poner en *Internet* sus metadatos, con una adopción del protocolo lenta y progresiva, elaborándose en el 2002 la versión 2.0 de dicho protocolo (Guajardo, 2010).

1.4.2.1 Estructura de OAI-PMH

Los metadatos a transmitir vía OAI-PMH son codificados en DC con el objetivo de minimizar los problemas derivados de las conversiones entre múltiples formatos. Usa transacciones mediante el Protocolo de Transferencia de Hipertexto (HTTP del inglés *Hypertext Transfer Protocol*) para emitir preguntas y obtener respuestas entre un servidor o archivo y un cliente o servicio recolector de metadatos y opera en conjunto con la tecnología Lenguaje de Marcado Extensible (XML del inglés *Extensible Markup Language*), dividiendo el trabajo entre proveedores de datos (DP del inglés *data provider*) y proveedores de servicios (SP del inglés *service provider*), basando la arquitectura de OAI-PMH en clientes y servidores.

Los DP son los propios repositorios y revistas que exponen los metadatos de sus contenidos. Los SP son aquellas herramientas desarrolladas para realizar peticiones a los DP, indexar contenidos e interpretar resultados (Fernández y col., 2011) para presentárselo a los usuarios finales.

Esta estructura de interoperabilidad corresponde al enfoque en la que los datos y metadatos son transferidos desde la fuente remota al destino en el cual se realizarán los servicios de búsqueda. Las peticiones se emiten utilizando los métodos GET o POST del protocolo HTTP y las respuestas de los proveedores de datos son documentos XML.

El protocolo soporta múltiples formatos para expresar los metadatos, no obstante requiere que todos los servidores ofrezcan los registros utilizando DC codificado en XML. Además de este formato cada servidor es libre de ofrecer los registros en otros formatos adicionales (MARC por ejemplo). Un cliente puede pedir que los registros se le sirvan en cualquiera de los formatos soportados por el servidor, esto con el fin de que en el futuro las diferentes comunidades que utilicen el protocolo definan sus propios formatos que sean más precisos que el DC. Por ejemplo la comunidad de archivos de *eprints* está trabajando en un formato denominado AMF (*Academic Metadata Format*) que describirá todos los

elementos que intervienen en el proceso de comunicación científica: documentos, autores, instituciones y canales de distribución de documentos (Barrueco, 2003).

Desde 2001 se ha mantenido constante el crecimiento del número de archivos que han implementado el protocolo OAI-PMH, hasta llegar a más de cuarenta y cinco archivos de variadas disciplinas. Se pueden destacar el CERN (informes y publicaciones en Física), Citebase y los archivos del área de Bibliotecología y Documentación, tales como @rchiveSIC, DLIST (*Digital Library of Information Science and Technology*) y E-LIS (*Eprints in Library and Information Science*) (Guajardo, 2010).

Igualmente los proveedores de servicios han crecido, siendo los más reconocidos a nivel mundial mencionados por la misma fuente:

- ARC¹.
- OAlster².
- Perseus³.
- Cyclades⁴.
- Scirus⁵.
- Hispana⁶.
- OpCit⁷.

Mientras que en América Latina, los más importantes mencionados por la misma fuente son:

- BDTD⁸.
- Latin American Open Archives Portal⁹.
- Cybertesis¹⁰.
- Scielo¹¹.
- Red de Bibliotecas Virtuales de Ciencias Sociales Clacso¹².

¹<http://dlib.cs.odu.edu/ARC.html/>

²<http://www.oclc.org/oaister/>

³<http://www.perseus.tufts.edu/hopper/>

⁴<http://www.ercim.eu/cyclades/>

⁵<http://www.scirus.com/>

⁶<http://roai.mcu.es/es/inicio/inicio.cmd/>

⁷<http://opcit.eprints.org/>

⁸<http://bdttd.ibict.br/>

⁹<http://lanic.utexas.edu/project/laoap/>

¹⁰<http://www.cybertesis.net/>

¹¹<http://www.scielo.org/php/index.php>

- Captura¹³.
- BDBComp¹⁴.

1.4.2.2 La comunicación OAI-PMH

En la comunicación OAI, las solicitudes o peticiones realizadas entre el SP y DP se realizan sobre la base de seis verbos, mediante los cuales un SP podrá recuperar la información que contiene el DP. Estos verbos mencionados por Fernández y col. (2011) son:

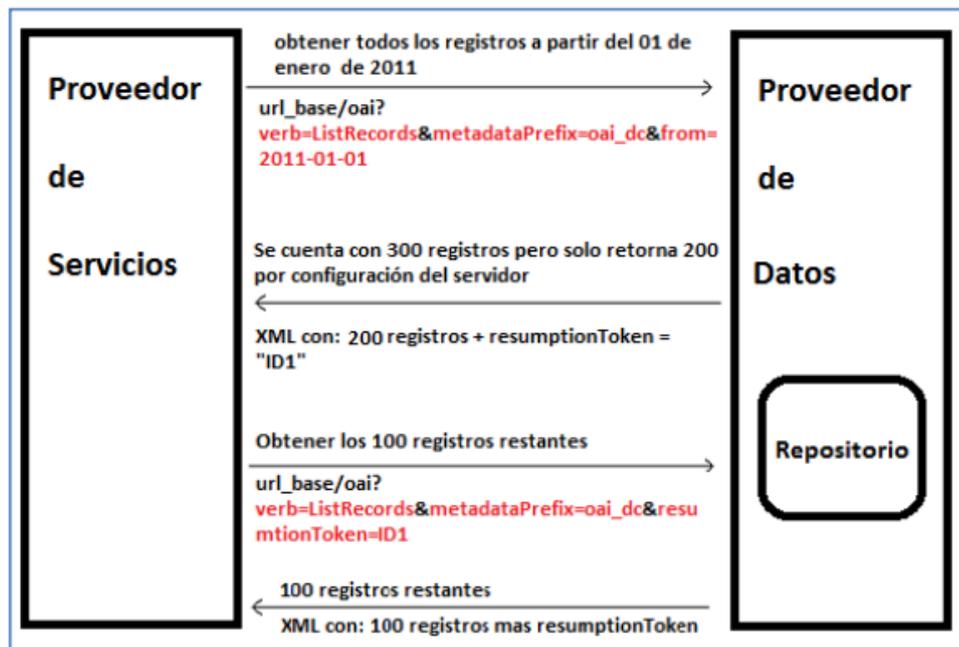
- **Identify:** pregunta por la identidad de una fuente, la respuesta da información del nombre del repositorio, de la granularidad de la fecha, de la fecha de inicio del repositorio y de la identidad en sentido general.
- **ListMetadataFormats:** pregunta por los estándares de metadatos con que están descritos los documentos en la fuente encuestada, la respuesta da una lista de los estándares de metadatos.
- **ListSets:** encuesta sobre la estructura del repositorio, la respuesta contiene la estructura temática del repositorio.
- **ListRecords:** pide el listado de los documentos, debe incluir la variable *metadataPrefix*, el valor asignado a esta variable es uno de los resultados del *ListMetadataFormats*.
- **ListIdentifier:** es una forma abreviada del *ListRecords*, pero la respuesta solo recupera los encabezados.
- **Getrecord:** pide un recurso específico, para ello usa la variable *identifier*, cuyo valor se obtiene del *ListIdentifier*, además la variable *metadataPrefix*.

A través de dichos verbos se establece la comunicación entre DP y SP, complementándose con otras variables como: *from* y *until* para una recopilación basada en fechas, *set* para recopilación basada en materias y *resumptionToken* para el control del flujo. En la figura 1.1 se muestra un ejemplo de cómo se implementa la comunicación entre los proveedores.

¹²<http://www.biblioteca.clacso.edu.ar/>

¹³<http://www.captura.uchile.cl/jspui/>

¹⁴<http://www.lbd.dcc.ufmg.br/bdbcomp/>



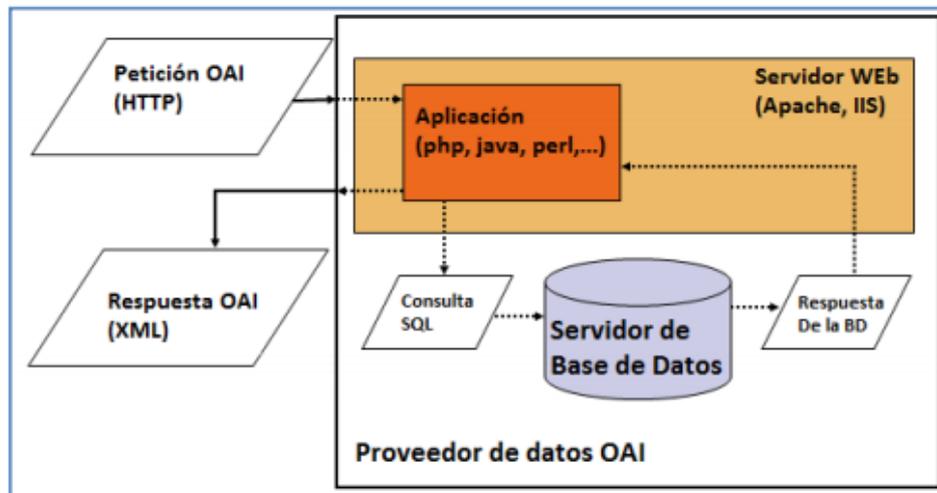
Fuente: Fernández y col. (2011). [Artículo en línea, 1 de Diciembre de 2011].

Fig. 1.1: Ejemplo de una comunicación entre un proveedor de servicios y un proveedor de datos.

1.4.2.3 Proveedor de datos

Para la implementación de un DP (Fig. 1.2) basta con respetar el estándar de metadatos (DublinCore) para la descripción del documento y generar un RDF (ver descripción en sección 1.7.3.3) para “especificar semántica a los datos basados en XML de una manera interoperable y estandarizada” (Abián, 2006). Todo esto se simplifica a la idea de generar un XML estándar, para las diferentes peticiones de los SP. Un proveedor de datos está compuesto por:

- Intérprete para validar las peticiones (puede estar implementado en cualquier lenguaje de programación)
- Generador de errores con respuestas XML.
- Interfaz de acceso a datos para extraer los metadatos.
- Generador XML para emitir las respuestas.
- Servidor Web.
- Servidor de Base de datos.



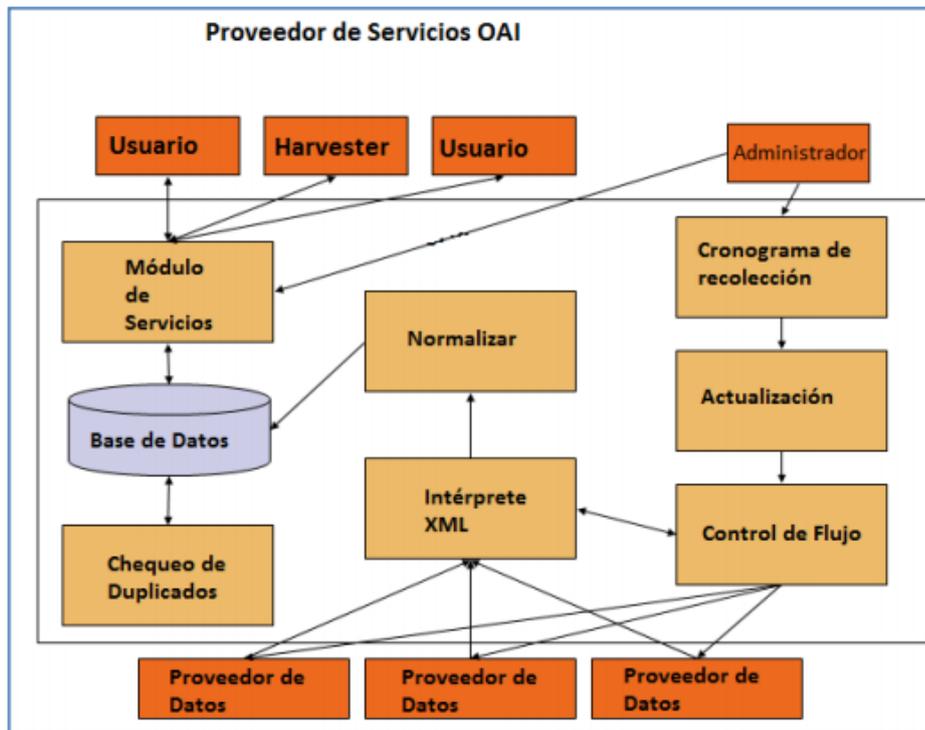
Fuente: Echebarría (2007). [Artículo en línea, 1 de Diciembre de 2011].

Fig. 1.2: Proveedor de datos.

1.4.2.4 Proveedor de servicios

Contrario a los DP, las instituciones prefieren desarrollar su propio SP (Fig. 1.3). Cuando existe un SP, se cuenta con una herramienta que permite indexar contenidos y proveer interfaces para la recuperación de información (Lagoze, 2005). Debe tener presente:

- La selección y validación de proveedores de datos.
- El control de flujo.
- La planificación de recolecciones.
- La recolección basada en que los datos y metadatos son transferidos desde la fuente remota al destino en el cual se realizarán los servicios de búsqueda.
- La detección de contenido duplicado.
- El análisis de la granularidad de las fechas.
- Interfaz de usuario para la interacción con el público.



Fuente: Echebarría (2007). [Artículo en línea, 1 de Diciembre de 2011].

Fig. 1.3: Proveedor de servicios OAI-PMH.

Como ejemplos de SP se pueden citar a Oaister, Citebase, Metalis, Roai y DRIVER.

1.5 Sistemas para repositorios

Desde los orígenes de OAI-PMH, han surgido una serie de herramientas que soportan diferentes aspectos, roles y funciones de la arquitectura del protocolo, con el fin de facilitar la distribución de información científica y académica y de cualquier tipo de contenido electrónico a través de la Web, dotándolos de un DP.

El desarrollo de un DP no es la solución más viable, sino hacer uso de las herramientas existentes. Las plataformas de aplicaciones para repositorios varían entre libre y comercial.

Según Bueno y Rodríguez (2007), entre las herramientas más exitosas para crear un repositorio documental y garantizar el proveedor de datos OAI haciendo uso de las tecnologías libres son:

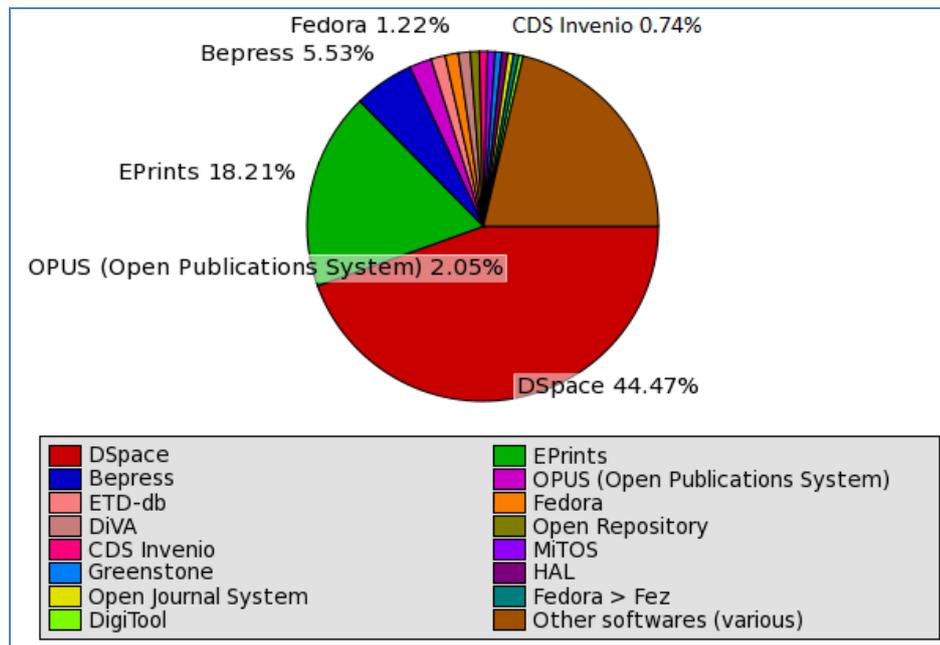
- DSpace: desarrollado conjuntamente por los laboratorios Hewlett-Packard y por la biblioteca del MIT (*Massachusetts Institute of Technology*) específicamente para constituir servicios de

repositorio institucional. DSpace es un sistema diseñado para la captura, almacenamiento, indexación, preservación y redistribución de la producción intelectual del personal investigador de la universidad en formato digital..

- GNU EPrints (antes EPrints): ha sido desarrollado en la Universidad de Southampton y actualmente está financiado por JISC (*Joint Information Systems Committee*) del Reino Unido, como parte del *Open Citation Project* y por la NSF (*Nacional Science Foundation*) de EEUU. En principio está diseñado para la creación de repositorios de *eprints*, aunque puede ser empleado con otros objetivos. Hasta el momento, GNU EPrints es el segundo sistema más utilizado en la creación de repositorios (superado por DSpace). Es muy empleado en la gestión del Preprints y Postprints.
- Fedora (*Flexible Extensible Digital Object and Repository Architecture*): ha sido elaborado conjuntamente por la Universidad de Virginia y la Universidad de Cornell (EEUU) con un proyecto que comenzó en 1998 con financiación de la *Fundación Andrew W. Mellon*. El proyecto Fedora pretende proporcionar repositorios digitales libres que puedan emplearse como fundamento para distintos tipos de sistemas de gestión de información: repositorios institucionales, bibliotecas digitales y otros sistemas de gestión de contenido, gestión de activos digitales, publicación académica y preservación digital.

Existen otros sistemas como Archimede, ARNO, CDSware, Mycore y OPUS. Otros sistemas usados que soportan el protocolo OAI-PMH y que no son precisamente sistemas para repositorios son: OJS (*Open Journal System*), OCS (*Open Conferences System*), *Keystone Digital Library Suit*, Greenstone y Alejandría.

Todos estos sistemas tienen la característica que pueden hacer de proveedores de datos y de servicios OAI.



Fuente: <http://roar.eprints.org/>. [Artículo en línea, 20 de Mayo de 2012].

Fig. 1.4: Uso de herramientas para repositorios.

1.6 Drupal para el desarrollo de sistemas de información tipo biblioteca

Para el desarrollo de un SI tipo biblioteca, existe un sin número de herramientas, entre ellos, los llamados Sistemas de Gestión de Contenidos (CMS del inglés *Content Management Systems*), que pueden resultar muy útiles en estos desarrollos. Drupal es un CMS muy empleado en proyectos de corte bibliotecario y se incluye como una posibilidad para implementar el protocolo OAI-PMH para su uso como un SP.

Drupal puede ser empleado para implantar un DP con facilidad. Para esto se requiere de la instalación de algunos módulos como el Content ConstructionKit (CCK) y algunos especializados para temas de bibliotecología como el módulo MARC, Bibliography y OAI2forCCK (Chalon, 2008). Que Drupal cuente con desarrollos para operar como un DP, también es importante que brinde la capacidad de funcionar como un SP, implementando así, el completo funcionamiento de OAI-PMH.

1.6.1 Drupal

Es un sistema de código abierto, con licencia GNU/GPL, escrito en PHP, de propósito general diseñado para emplearse en cualquier tipo de proyecto y es desarrollado y mantenido por una activa

comunidad de usuarios. Destaca, por la calidad de su código y de las páginas generadas, el respeto de los estándares de la Web y un énfasis especial en la usabilidad y consistencia de todo el sistema. Drupal es un sistema multiusuario, multiplataforma, multilenguaje, multipropósito, extensible, modular y muy configurable. es un sistema dinámico: en lugar de almacenar sus contenidos en archivos estáticos en el sistema de ficheros del servidor de forma fija, el contenido textual de las páginas y otras configuraciones son almacenados en una base de datos y se editan utilizando un entorno Web (Egiguren, 2010).

Drupal ofrece prestaciones y funcionalidades que le permiten ser utilizado como plataforma de gestión de información digital en bibliotecas, archivos y servicios de información y documentación; por lo tanto, es muy útil que cuente con desarrollos que implementen los protocolos OAI-PMH y Z39.50 y sus respectivos estándares para que puedan ser usados en el desarrollo de SI, siguiendo las tendencias actuales de las bibliotecas digitales.

Según Tramullas (2010) las bibliotecas no han sido ajenas al enorme potencial de personalización y adecuación de Drupal para comunidades y servicios específicos y su desarrollo. Desde hace unos años es creciente el número de bibliotecas que fundamentan su presencia y servicios en *Internet* a través de Drupal.

La importancia que Drupal está alcanzando en el entorno bibliotecario justifica que, desde hace unos años, se celebren reuniones y talleres sectoriales, conocidos como Drupal4Lib, en el marco de los congresos que celebra la *American Library Association*. También Drupal ha sido objeto de interés en las conferencias especializadas Code4Lib, que se celebran desde 2006. Se han incluido actividades sobre Drupal en *Computers for Libraries 2008* y en la conferencia *Access 2008*.

1.7 Entorno tecnológico para el trabajo con Drupal

1.7.1 Sistema de Gestión de Base de Datos

Un sistema gestor de base de datos se define como los programas que permiten crear, administrar y gestionar de manera clara y sencilla la información contenida en una base de datos (Márquez, 2009).

Una base de datos es un conjunto de datos organizados y relacionados entre sí, los cuales son almacenados sistemáticamente para su uso posterior. Por lo tanto se puede considerar a una

biblioteca como una base de datos formada por documentos y papeles impresos e indexados para su consulta.

1.7.1.1 MySQL

Es un Sistema de Gestión de Base de Datos (SGBD) relacional, su fuerte liga con PHP. Líderes de la industria como Yahoo, Google, Nokia y Youtube lo utilizan. Desde abril de 2009 la compañía MySQL AB fue vendida a *Oracle Corporation*, esto provocó que MySQL pasara a ser propietaria. Se ofrece bajo licencia GNU GPL para cualquier uso compatible con esta licencia. Mientras que aquellas empresas que quieran incorporarlo en productos privativos deben de comprar una licencia que les permita el uso. Cuantiosos CMS como Drupal, Joomla y phpBB usan MySQL (Oracle, 2011).

1.7.1.2 PostgreSQL

Es un SGBD objeto-relacional, basado en el proyecto POSTGRES, de la universidad de Berkeley, distribuido bajo licencia BSD y con su código fuente disponible libremente. El director de este proyecto es el profesor Michael Stonebraker y fue patrocinado por *Defense Advanced Research Projects Agency (DARPA)*, el *Army Research Office (ARO)*, el *National Science Foundation (NSF)* y ESL, Inc. Utiliza un modelo cliente/servidor y usa multiprocesos en vez de multihilos para garantizar la estabilidad del sistema. Un fallo en uno de los procesos no afectará el resto y el sistema continuará funcionando. PostgreSQL funciona muy bien con grandes cantidades de datos y una alta concurrencia de usuarios accediendo a la vez al sistema (Postgresql, 2011). Contiene características de la orientación a objetos, como puede ser la herencia, tipos de datos, funciones, restricciones, disparadores, reglas e integridad transaccional, sin embargo, no es un sistema de gestión de bases de datos puramente orientado a objetos (Redondo, 2011).

1.7.2 Servidor Web Apache

Es un servidor altamente configurable de diseño modular libre y de código abierto. Es multiplataforma. Posee una alta integración con diversas tecnologías, lenguajes, plataformas, bases de datos, entre otros. Opera con lenguajes como Java, Perl y PHP. Tiene capacidad para servir páginas tanto de contenido estático, como de contenido dinámico a través de otras herramientas soportadas que facilitan la actualización de los contenidos mediante bases de datos, ficheros u otras fuentes de información. Permite personalizar la respuesta ante posibles errores que se puedan presentar en el servidor. Posee una alta configurabilidad en la creación y gestión de logs (Apache, 2010). Apache tiene

amplia aceptación en la red: desde 1996, Apache, es el servidor HTTP más usado. Alcanzó su máxima cuota de mercado en 2005 siendo el servidor empleado en el 70% de los sitios Web en el mundo, sin embargo ha sufrido un descenso en su cuota de mercado en los últimos años. (Estadísticas históricas y de uso diario proporcionadas por Netcraft, 2012)

1.7.3 Programación para la Web

Un lenguaje de programación es un sistema de escritura que da lugar a expresar un conjunto de instrucciones que controlen el funcionamiento de una computadora. Los lenguajes de programación orientados a la Web están divididos en dos grupos, los del lado del cliente y del lado del servidor.

1.7.3.1 PHP

Procesador de Hipertextos (PHP del inglés *Hypertext Pre-processor*) es un lenguaje interpretado de alto nivel embebido en páginas HTML y ejecutado en el servidor. Con PHP se puede hacer numerosas cosas, como por ejemplo, procesar información en formularios, foros de discusión y páginas dinámicas. También ofrece la integración con varias bibliotecas externas, que permiten al desarrollador desde generar documentos PDF hasta analizar código XML. PHP soporta diversas bases de datos como MySQL, Informix, Oracle, Sybase, Solid, PostgreSQL, Generic y ODBC. Debido a su amplia distribución PHP está perfectamente soportado por una gran comunidad de desarrolladores. (w3schools, 2011).

Ventajas de PHP según w3schools (2012):

- PHP corre en casi cualquier plataforma utilizando el mismo código fuente.
- La sintaxis de PHP es similar a la de C, por esto cualquiera con experiencia en lenguaje del estilo C podrá entender rápidamente PHP.
- Capacidad de conexión con la mayoría de los manejadores de base de datos que se utilizan actualmente, como por ejemplo MS SQL, Oracle, MySQL y PostgreSQL.

1.7.3.2 XML

El desarrollo de XML comenzó en 1996 y desde entonces ha tenido un desarrollo exponencial. El desarrollo de este metalenguaje es una iniciativa del *World Wide Web Consortium (W3C)* y su principal

característica es presentar una mayor flexibilidad que el HTML y un uso más apropiado en la Web ya que presenta una mayor facilidad para recuperar información al facilitar el marcado según el contenido semántico del documento; los argumentos a favor de este esquema se basan en la sintaxis, el hecho de estar orientado a objetos, el polimorfismo y la posibilidad de definir nuevos tipos de datos para refinar el sistema de marcado o para estructurar de forma más coherente la información (BNC, 2009).

1.7.3.3 RDF

El Marco de Descripción de Recursos (del inglés, *Resource Description Framework*) es un lenguaje para representar información sobre recursos en la Web desarrollado por la iniciativa W3C. Permite la interoperabilidad entre aplicaciones que intercambian información comprensible vía Web, para proporcionar una infraestructura que soporte actividades de metadatos. Está particularmente indicado para representar metadatos sobre recursos Web, tales como el título, autor, modificaciones de los datos de la página, *copyright* y otras licencias de información sobre documentos en línea, así como la disponibilidad para algunos recursos compartidos. Se trata de un modelo de datos para objetos (recursos) (Lamarca, 2006).

1.7.3.4 XHTML

Lenguaje de Marcado de Hipertexto Extensible (del inglés, *Extensible Hypertext Markup Language*) es una versión más limpia y estricta de HTML, que nace precisamente con el objetivo de reemplazar a HTML ante su limitación de uso con las cada vez más abundantes herramientas basadas en XML. XHTML extiende HTML 4.0 combinando la sintaxis de HTML, diseñado para mostrar datos, con la de XML, diseñado para describir los datos. Surge como el lenguaje cuyo etiquetado, más estricto que HTML, permite una correcta interpretación de la información independientemente del dispositivo desde el que se accede a ella (W3C, 2008).

1.7.3.5 CSS

Hojas de Estilos en Cascada (del inglés *Cascading Style Sheets*) es un mecanismo simple que describe cómo se va a mostrar un documento en la pantalla, o cómo se va a imprimir, o incluso cómo va a ser pronunciada la información presente en ese documento a través de un dispositivo de lectura. Esta forma de descripción de estilos ofrece a los desarrolladores el control total sobre estilo y formato de sus documentos. CSS se utiliza para dar estilo a documentos HTML y XML, separando el contenido de la presentación (W3C, 2005).

1.7.3.6 JavaScript

Es un lenguaje de script, desarrollado por la empresa Netscape en los años de 1990 con su primera versión de Livescript para incrementar las funcionalidades del lenguaje HTML, brindándole un mayor dinamismo a las páginas Web. Es un lenguaje que puede ser utilizado por profesionales y para quienes se inician en el desarrollo y diseño de sitios. No requiere de compilación ya que el lenguaje funciona del lado del cliente, donde los navegadores son los encargados de interpretar estos códigos (Pérez, 2007).

1.7.3.7 JQuery

Es una biblioteca de JavaScript, creada inicialmente por John Resig, que permite simplificar la manera de interactuar con los documentos HTML, manipular el árbol DOM, manejar eventos, desarrollar animaciones y agregar interacción con la técnica AJAX. Permite selección de elementos DOM, interactividad y modificaciones del árbol DOM, eventos, manipulación de la hoja de estilos CSS, efectos y animaciones, animaciones personalizadas, AJAX, utilidades varias como obtener información del navegador, operar con objetos y vectores.

1.7.3.8 Ajax

El término AJAX se acuñó por primera vez en el artículo “*Ajax: A New Approach to Web Applications*” publicado por Jesse James Garrett el 18 de Febrero de 2005. Es un acrónimo de Asynchronous JavaScript + XML, que se puede traducir como “JavaScript asíncrono + XML”. Permite mejorar completamente la interacción del usuario con la aplicación, evitando las recargas constantes de la página, ya que el intercambio de información con el servidor se produce en un segundo plano. Las peticiones HTTP al servidor se transforman en peticiones JavaScript que se realizan al elemento encargado de AJAX.

“Ajax no es una tecnología en sí mismo. En realidad, se trata de la unión de varias tecnologías que se desarrollan de forma autónoma y que se unen de formas nuevas y sorprendentes” (Eguiluz, 2009).

1.7.4 Proceso de mejora de software

Los Procesos de Mejora constituyen una alternativa para elevar la Calidad del Software.

1.7.4.1 Proceso

Se puede definir como un “conjunto coordinado de actividades que proporcionan un valor añadido al cliente (ciudadano o usuario interno), entregándole un resultado (el producto o servicio que se trate) que le satisfaga, partiendo de una serie de entradas al proceso y con la utilización de una serie de recursos” (Merino y col., 2003).

Un proceso es un conjunto de actividades o tareas que se interrelacionan y que se ejecutan siguiendo un orden lógico con el propósito de alcanzar un resultado específico a partir de las entradas de recursos e información.

1.7.4.2 Mejora

Está vigente en diferentes esferas de la vida, por lo general en ambientes empresariales donde ha dejado de ser una opción para convertirse en una imperiosa necesidad.

Mejora, proviene del verbo mejorar que implica cambiar, El Diccionario Enciclopédico (2001) define mejora como: “Acción o efecto de mejorar” y “Cambio hecho en una cosa, por lo que resulta mejorada”.

Según la RAE, mejora es: “Medra, adelantamiento o mejora de algo”.

1.7.4.3 CMMI

La mejora del “Proceso Software” es un mecanismo de mejora continua de la calidad. Se puede utilizar para elevar capacidad de los contratistas, para auditar desarrollos internos y para planificar la estrategia de ingeniería de las aplicaciones de la empresa (Armas y col., 2007).

Uno de los modelos de mejora de procesos más usados en las organizaciones es el Modelo de Capacidad y Madurez Integrado (CMMI, del inglés *Capability Maturity Model Integration*). Este es un modelo que ayuda a integrar las funciones de la organización, conducir la mejora de los procesos, proporciona una guía de calidad de los procesos y puntos de referencia para la evaluación de estos. El propósito del modelo es proveer una guía para mejorar los procesos de la organización y la capacidad para gestionar el desarrollo, la adquisición y el mantenimiento de productos y servicios (López y col., 2007).

1.7.5 Metodologías de desarrollo

Se define como metodología al conjunto de estrategias, métodos o actividades intencionadas, organizadas, secuenciadas e integradas, que permitan el logro de aprendizajes significativos y de calidad (Lagos, 2008).

A su vez, una metodología para el desarrollo del proceso de software, es el conjunto de actividades necesarias para transformar los requisitos de los usuarios en un sistema, definiendo: Quién debe hacer Qué, Cuándo y Cómo debe hacerlo. Dentro de las metodologías más conocidas se hallan: Proceso Unificado de Desarrollo (RUP del inglés *Rational Unified Process*), Programación Extrema (XP del inglés *eXtreme Programming*) y *Microsoft Solution Framework* (MSF).

1.7.5.1 RUP

Es un proceso de desarrollo de aplicaciones informáticas que junto con el Lenguaje Unificado de Modelado (UML del inglés *Unified Modeling Language*), se emplean para implementar y documentar sistemas orientados a objetos. El mismo está dirigido por casos de uso, centrado en la arquitectura e iterativo e incremental.

A partir de la propia fuente, RUP formula el desarrollo de productos en cuatro fases:

1. Inicio
2. Elaboración
3. Construcción
4. Transición

En cada fase se ejecutan una o varias iteraciones, en dependencia del tamaño del proyecto. Propone nueve flujos de trabajo que son desarrollados en cada una de sus cuatro fases, donde los seis primeros son ingenieriles y el resto de soporte. Ellos son:

1. Modelado del negocio
2. Análisis de requisitos
3. Análisis y diseño
4. Implementación
5. Pruebas
6. Despliegue

7. Gestión de configuración y cambio
8. Gestión del proyecto
9. Gestión del entorno

El proceso define una serie de roles que se distribuyen entre los miembros del proyecto y que definen las tareas de cada uno y el resultado que de ellos se espera. El proceso de desarrollo RUP aplica varias de las mejores prácticas en el desarrollo moderno de sistemas informáticos en una forma que se adapta a un amplio rango de proyectos y de organizaciones. Provee a cada miembro del equipo, un fácil acceso a una base de conocimiento con guías, plantillas y herramientas para todas las actividades críticas del desarrollo. Esta metodología permite que todos los integrantes de equipo de trabajo, conozcan y compartan el proceso de desarrollo, una base de conocimientos y los distintos modelos de cómo desarrollar utilizando un lenguaje de modelado común: UML.

1.7.6 Herramientas CASE

Conjunto de programas que dan asistencia a los analistas, ingenieros de Software y desarrolladores, durante todo el ciclo de vida de desarrollo de un proyecto informático. Este puede ser generalmente aplicado a cualquier sistema o colección de herramientas que ayudan a automatizar el proceso de diseño y desarrollo de sistemas.

1.7.6.1 NetBeans IDE

Herramienta libre, de código abierto que presenta un entorno de desarrollo integrado para desarrolladores. Presenta todas las herramientas necesarias para crear aplicaciones profesionales de escritorio, empresariales, Web y aplicaciones móviles con la plataforma Java, así como PHP, JavaScript y otros. (NetBeans, 2012). Es un entorno de desarrollo integrado, el cual, permite que las aplicaciones sean desarrolladas con un conjunto de componentes denominados módulos. Las aplicaciones realizadas con módulos, se pueden agrandar añadiendo más módulos, ya que pueden ser independientes los unos de los otros (Redondo, 2011).

1.7.6.2 Visual Paradigm

Herramienta de diseño UML y herramienta CASE UML diseñado para apoyar al desarrollo de sistemas dentro de la rama de la Ciencias de la Computación. Soporta estándares tales como UML, SysML, BPMN, XMI, etc. Ofrece un conjunto completo de herramientas de desarrollo necesario para la captura

de requisitos, la planificación y pruebas, el modelado de clases y datos, etcétera (Visual Paradigm, 2011).

1.8 Casos de estudio de bibliotecas desarrollados con Drupal

En la gran red de redes se encuentran numerosos proyectos bibliotecarios desarrollados con el CMS Drupal, siendo esta, una herramienta muy potente para el desarrollo de bibliotecas y archivos, entre los que se destacan según Tramullas (2010) RecBib: Recursos Bibliotecarios, ARTIUM: Biblioteca y Centro de Documentación y Bibliotécnica 2.0: Biblioteca Digital de la Universidad Politécnica de Cataluña, Recida.net: Portal Web de la red de centros de información y documentación ambiental y Tabula Decimal: Laboratorio virtual de aplicaciones para la gestión de contenidos digitales.

- RecBib: en un principio, el sitio Web de la misma era estático y desarrollado con FrontPage y luego con Dreamweaver. Nació con el fin de recopilar una gran colección de recursos útiles, dentro del campo de la Biblioteconomía y la Documentación. Posteriormente la cantidad de usuarios registrados comenzaba a crecer y la necesidad de brindar mejores servicios y de contar con un sitio dinámico era imprescindible, por lo tanto se migró a Drupal.
- ARTIUM: el Centro de Documentación de ARTIUM pretende potenciar la creación y difusión de contenidos y servicios digitales dentro del marco de la cultura y el arte. Entre sus objetivos se encuentran compartir y distribuir información de forma eficaz para la gestión del capital intelectual del museo contemporáneo, disponer de un sistema flexible y eficiente de búsqueda, indexación y consulta de la información procesada y que permita la publicación de documentos en múltiples formatos: doc, pdf, html, etcétera.
- Bibliotécnica 2.0: su origen se remonta al año 1994. Se abrió un proceso reflexivo respecto a una reorganización de los contenidos y servicios, así como que la gestión de estos se pudiera realizar de forma completamente descentralizada y adaptada a las necesidades y características de cada centro, teniendo en cuenta aspectos como la posibilidad de crear flujos de trabajo en el proceso de creación y edición de contenidos recogidos en una biblioteca digital.

1.9 Propuesta de solución tecnológica

En el desarrollo de la investigación se llega a la conclusión que se usarán para la ejecución del sistema las siguientes tecnologías:

- Uso del protocolo OAI-PMH que es ampliamente implementado en sistemas de tipo biblioteca para distribuir la información en *Internet* de forma libre.
- Se toma como referencia el estándar de metadatos DC para la descripción de los documentos que está estandarizado en el movimiento OA y garantiza la descripción de documentos de carácter científico y es usado por sistemas como Redalyc (Red de Revistas Científicas de América Latina y el Caribe, España y Portugal), Calames OAI server (Biblioteca de investigación) Scirus, OAlster, Red de Bibliotecas Virtuales de Ciencias Sociales Clacso, Scielo, Latin American Open Archives Portal y el Portal de la Biblioteca UCI.
- La tecnología Web del lado del cliente XHTML 1.0 y CSS 2.1.
- El lenguaje dinámico del lado del servidor PHP 5.3.6.
- El CMS Drupal 6.26 por su amplio uso en proyectos bibliotecarios y capacidad para funcionar como un SP. El SP puede ser ejecutado en cualquiera de las versiones de Drupal seis; para esto fue probado en el 6.26, 6.22, 6.20 y 6.19. Ejemplos del empleo de este CMS se encuentran en proyectos como RecBib, ARTIUM, Bibliotécnica 2.0, Recida.net, Tabula Decimal y Biblioteca Digital Alma Mater.
- El SGBD PostgreSQL 9.1.3 por su gran escalabilidad y es capaz de ajustarse al número de CPUs y a la cantidad de memoria que posee el sistema de forma óptima haciéndolo capaz de soportar una gran cantidad de peticiones simultáneas de manera correcta.
- El Servidor Web Apache 2.2.20.
- Las herramientas NetBeans IDE 7.1 para la programación y Visual Paradigm 8.0 para el modelado de diagramas.
- La metodología RUP con uso del proceso de mejora de software con nivel dos de CMMI.

Capítulo 2

PROPUESTA DE SOLUCIÓN DEL PROVEEDOR DE SERVICIOS OAI PARA EL SISTEMA DE GESTIÓN DE CONTENIDOS DRUPAL

En el presente capítulo se realiza un análisis y se detalla la propuesta de solución del sistema. Para ello se describen las reglas del negocio asociados al objeto de estudio y de automatización. Se da a conocer una propuesta del sistema y luego se muestra una descripción del modelo de dominio, que facilita el paso a los requerimientos del sistema y la identificación de los actores que interactúan con el mismo. También se analizan los requerimientos funcionales y no funcionales.

2.1 Objeto de automatización

Se pretende automatizar la adquisición de documentos desde fuentes externas (bibliotecas, repositorios, revistas, etc.) que se distribuyan bajo las normas del protocolo Iniciativa de Archivos Abiertos - Protocolo para la Transmisión de Metadatos (OAI-PMH por sus siglas en inglés). Además configurar el cronograma de indexación y los verbos usados, adaptados a la arquitectura del Sistema de Gestión de Contenidos (CMS por sus siglas en inglés) Drupal. Este fenómeno tiene lugar mediante la existencia de un proveedor de datos (DP, del inglés *data provider*) y un proveedor de servicios (SP, del inglés *service provider*).

2.2 Información que se maneja

En la aplicación se manejan los documentos basados en el estándar de metadatos *Dublin Core* (DC), que propone quince campos para la descripción de estos recursos. También se operan con los recursos y sus fuentes correspondientes, roles de usuarios y permisos para las diferentes acciones del indexador. Las relaciones entre documentos y nodos en la arquitectura de Drupal, proporcionando un tratamiento de clasificación a los diferentes campos del estándar de metadatos.

2.3 Propuesta del sistema

El sistema realizará la indexación de documentos desde proveedores de datos OAI, basándose en peticiones OAI. Primeramente se hace una petición sobre el verbo *Identify* para obtener los datos de la fuente y a continuación una petición *ListMetadataFormats* para obtener el estándar de metadatos que

emplea la fuente, evaluando si esta es correcta y aceptada, atendiendo a que responde y que emplea `oai_dc` como formato de metadatos.

Si la fuente es correcta y contiene colecciones, se almacenan las colecciones de preferencia mediante el verbo `ListSets` con el objetivo de mantener bajo una clasificación determinada a los recursos provenientes de la fuente. También se efectúan varias peticiones `Listrecords` con el uso del `MetadataPrefix` y `resumptionToken` para obtener los documentos que son transformados en nodos de Drupal y almacenados (los metadatos de estos documentos responden al estándar DC, ver en sección [1.3.2](#), quedando disponibles para que los usuarios puedan recuperarlos.

También se recopilan las fuentes de datos y las fechas de las últimas actualizaciones para emplearlas en próximos procesos de indexación. Todas estas acciones las realiza un usuario con los permisos necesarios. Se gestionan los roles de usuarios y sus permisos para gestionar todo el proceso del SP. El sistema también genera un `schedule` de actualización, para garantizar la indexación automática mediante tareas cron.

Otras acciones del sistema están relacionadas con la gestión de los documentos y de los orígenes de datos, los procesos de modificar y eliminar. También se trabaja en la representación del contenido, usando un modelo de clasificaciones para campos como el tipo, la fuente y las palabras claves.

2.4 Modelo de dominio

El modelo de dominio es una representación visual estática del entorno real objeto del proyecto. Se dice que es estática porque no representa la interacción en el tiempo de los objetos, sino que representa una visión "parada" de las clases y sus interacciones. Ayuda a comprender los conceptos que utilizan los usuarios, los conceptos con los que trabajan y con los que deberá trabajar la aplicación.

Captura los elementos más importantes de objetos en el contexto del sistema. Los objetos del dominio representan las “cosas” que existen o los eventos que suceden en el entorno en el que trabaja el sistema. Muchos de los objetos o clases pueden obtenerse de una especificación de requisitos o mediante la entrevista con los expertos del dominio. Las clases del dominio aparecen en tres formas características:

- Objetos del negocio que representan cosas que se manipulan en el negocio, como pedidos, cuentas y contratos.

- Objetos del mundo real y conceptos de los que el sistema debe hacer un seguimiento, como la aviación enemiga, misiles y trayectorias.
- Sucesos que ocurrirán o han ocurrido, como la llegada de un avión, su salida y la hora de la comida.

El modelo de dominio se describe mediante diagramas UML (especialmente mediante diagramas de clases).

Estos diagramas muestran a los clientes, usuarios, revisores y a otros desarrolladores las clases del dominio y cómo se relacionan unas con otras mediante asociaciones. El objetivo del modelo de dominio es comprender y describir las clases más importantes dentro del contexto del sistema.

2.4.1 Diagrama de clases del modelo del dominio

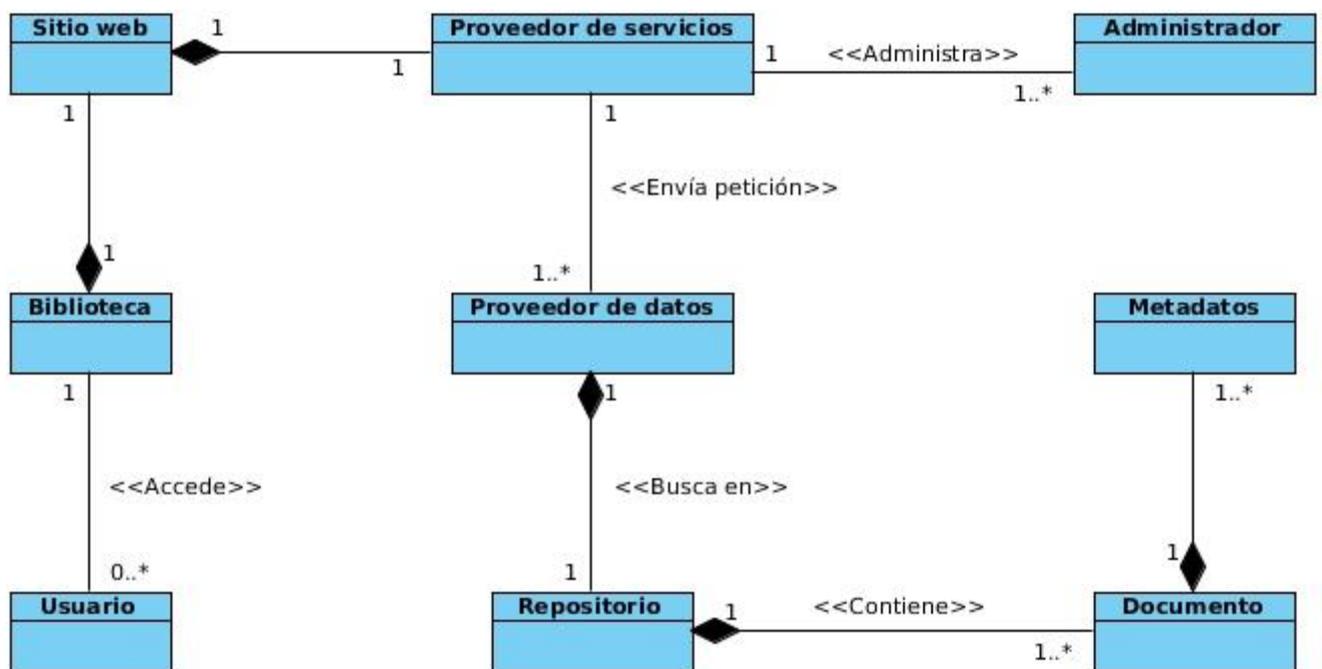


Fig. 2.1: Diagrama del modelo del dominio.

2.4.1.2 Descripción de las clases del modelo del dominio

Para comprender el modelado es necesario conocer el significado de los siguientes aspectos:

Administrador: persona que cuenta con permisos de administración sobre el sitio Web.

Biblioteca: entidad que hace uso de la aplicación para exponer y recuperar contenidos.

Usuario: persona que accede al sistema.

Documento: contiene la información científica que se les expondrá a los clientes.

Metadatos: información descriptiva sobre el contexto, calidad, condición o características de un recurso, dato u objeto que tiene la finalidad de facilitar su recuperación.

Proveedor de datos: mecanismo encargado de la recolección de metadatos a partir de las peticiones generadas por el SP.

Proveedor de servicios: herramienta encargada de realizar peticiones al DP para recuperar información bibliográfica y procesarla en el sistema donde se indexó.

Repositorios: archivos o depósitos en sitios centralizados donde se almacena y mantiene la información digital que será consumida por el SP.

Sitio Web: aplicación publicada en la gran red de redes de la biblioteca que gestiona los contenidos.

2.5 Especificación de requisitos de software

Un requisito de software es la condición o capacidad que tiene que ser alcanzada o poseída por un sistema o componente de un sistema para satisfacer un contrato, estándar, u otro documento impuesto formalmente.

2.5.1 Descripción de los actores

Para el correcto funcionamiento del sistema existirán dos actores: el Administrador y el Reloj.

Tabla 2.1. Actores del sistema.

Actor	Objetivo
Administrador	Persona encargada de gestionar documentos, el proceso de indexado y asignar roles y permisos.
Reloj	Personifica al encargado de indexar documentos cada un tiempo determinado.

2.5.2 Requisitos funcionales

Son aquellos que responden a: ¿Qué debe hacer el sistema? y describen las capacidades o condiciones que el sistema debe cumplir. En la tabla 2.2 se listan los requisitos funcionales que el sistema debe cumplir.

Tabla 2.2. *Requisitos funcionales.*

RF1. Autenticar	RF7. Listar fuente	RF13. Eliminar colección
RF2. Crear documento	RF8. Modificar fuente	RF14. Indexar documento manualmente
RF3. Listar documento	RF9. Eliminar fuente	RF15. Indexar documento automáticamente
RF4. Modificar documento	RF10. Crear colección	RF16. Actualizar token
RF5. Eliminar documento	RF11. Listar colección	RF17. Indexar documento desde archivo
RF6. Crear fuente	RF12. Modificar colección	

2.5.3 Requisitos no funcionales

Los requisitos no funcionales son las cualidades o propiedades que el producto debe tener. Debe pensarse en estas propiedades como las características que hacen al producto atractivo, usable, rápido o confiable. Determinan cómo tiene que comportarse el sistema, qué cualidades debe tener o cuán rápido o grande debe ser. El sistema presenta los siguientes requisitos no funcionales:

Usabilidad:

- RnF1. El usuario con rol de administrador podrá operar con las funcionalidades del proveedor de servicios.
- RnF2. Uso de paginados para representar la cantidad de documentos producto de las indexaciones.
- RnF3. Funcionalidades bien estructuradas para su fácil manejo.

Ambiente

- RnF4. Servidor Web Apache (versión 2.2.20).

- RnF5. Intérprete de aplicaciones Web (Mozilla Firefox 11.0, Internet Explorer 8, Google Chrome 17.0.963.46).
- RnF6. Intérprete de Procesador de Hipertextos (PHP del inglés *Hypertext Pre-processor*).
- RnF7. Sistema Gestor de Bases de Datos PostgreSQL 9.1. Para el funcionamiento del sistema puede ser empleado PostgreSQL o MySQL.
- RnF8. 512 MB de memoria de acceso aleatorio (RAM del inglés *random-access memory*) para el cliente.
- RnF9. Procesador Pentium IV.
- RnF10. Intel Core 2 Duo o equivalente para el servidor.
- RnF11. 4 GB de memoria RAM por parte del servidor.

Fiabilidad:

- RnF12. El tiempo medio de corrección de errores no debe exceder las 72 horas.
- RnF13. Los errores no deben estar por encima de dos.
- RnF14. Se debe contar con un sistema de salvadas externas para la información que maneja el sistema.

Soporte:

- RnF15. El sistema debe dar la posibilidad de ser mejorado, así como de incorporarle nuevos servicios en caso de ser necesarios.

Restricciones de diseño:

- RnF16. Lenguaje de programación PHP 5.3.6.
- RnF17. Lenguaje de Marcado de Hipertexto Extensible (XHTML del inglés *eXtensible HyperText Markup Language*).
- RnF18. Hojas de Estilos en Cascada (CSS del inglés *Cascading Style Sheets*).

- RnF19. Netbeans IDE 7.1.
- RnF20. CMS Drupal 6.x.

Interfaz:

Interfaces de usuario:

- RnF21. Diseño sencillo, que permita un rápido aprendizaje por parte del usuario.

Interfaces de Hardware:

- RnF22. Todos los servidores en los que estén desplegados los sistemas deben estar en un local que cuente con la climatización y el nivel de acceso adecuado para un sistema de esta índole.

Interfaces de comunicación:

- RnF23. La comunicación entre el cliente y el servidor Web será realizado a través del Protocolo de Transferencia de Hipertexto (HTTP del inglés *Hypertext Transfer Protocol*).
- RnF24. La comunicación entre el servidor Web y el servidor de Base de datos será realizado a través del Protocolo de Control de Transmisión / Protocolo de Internet (TCP/IP del inglés *Transmission Control Protocol / Internet Protocol*).
- RnF25. Las peticiones realizadas por el SP al DP serán mediante el protocolo HTTP.

Licencia:

- RnF26. Uso de la Licencia Pública General de GNU (GNU/GPL del inglés *GNU General Public License*) para el CMS Drupal.
- RnF27. Uso de la Distribución de Software Berkeley (BSD del inglés *Berkeley Software Distribution*) de PostgreSQL.
- RnF28. Uso de la licencia Apache Software (Licencia para versiones anteriores a 2.0).
- RnF29. Uso de la licencia *PHP License*.

Estándares aplicables:

- RnF30. Se sigue el estándar de programación sugerido por la arquitectura de Drupal.
- RnF31: Empleo del patrón de diseño GoF (*Gang of Four*, en español Pandilla de los Cuatro, formada por Erich Gamma, Richard Helm, Ralph Johnson y John Vlissides).
- RnF32. El nombre de cada una de las funciones que se definan debe estar escrito de la siguiente forma: nombre_del_modulo_nombre_de_la_funcion.
- RnF33. Cada sentencia condicional debe llevar sus respectivas llaves.
- RnF34. Los arreglos deben ser formateados dejando un espacio entre cada elemento, por ejemplo: `$array = array ('llave1'=>'nombre1', 'llave2'=>'nombre2');`
- RnF35. En los ficheros .module no se cierra el tag de PHP `?>`.
- RnF36. El Consorcio de la Web o Telaraña Mundial (W3C del inglés *World Wide Web Consortium*) es el encargado de formular la especificación de las hojas de estilo que servirán de estándar para los navegadores. Esta empresa ha definido hasta la fecha tres versiones de CSS, dentro de las que se mencionan: CSS 1 (1996), CSS 2 (1998) y CSS 3 (1999), encontrándose en estado de desarrollo.

Seguridad:

- RnF37. Controlar los permisos a nivel de roles.
- RnF38. Impedir el uso de PHP a cualquier usuario.

Portabilidad:

- RnF39. El sistema será multiplataforma, permitiendo ser montado en sistemas operativos basados en GNU/Linux y Windows.

2.6 Descripción del sistema propuesto

2.6.1 Listado de casos de uso del sistema

Tabla 2.3. Caso de uso Autenticar.

CU-1	Autenticar
Actor	Administrador
Descripción	Para lograr una interacción entre el administrador y el sistema, el administrador tiene que autenticarse, entrando el usuario y la contraseña.
Referencia	RF1

Tabla 2.4. Caso de uso Crear documento.

CU-2	Crear documento
Actor	Administrador
Descripción	Se crea un documento introduciendo todos los datos necesarios para esta operación.
Referencia	RF2

Tabla 2.5. Caso de uso Listar documento.

CU-3	Listar documento
Actor	Administrador
Descripción	El usuario con rol de administrador lista los documentos, una vez autenticado en el sistema.
Referencia	RF3

Tabla 2.6. Caso de uso Modificar documento.

CU-4	Modificar documento
Actor	Administrador
Descripción	Quien se comporta como administrador puede modificar los documentos

	que han sido almacenados en el sistema, una vez autenticados.
Referencia	RF4

Tabla 2.7. Caso de uso Eliminar documento.

CU-5	Eliminar documento
Actor	Administrador
Descripción	El administrador del sistema puede eliminar documentos que hayan sido creados en el sistema, luego de su autenticación.
Referencia	RF5

Tabla 2.8. Caso de uso Gestionar fuente.

CU-6	Gestionar fuente
Actor	Administrador
Descripción	El administrador una vez autenticado, puede crear, mostrar, modificar y eliminar fuentes que son empleadas para guardar las colecciones.
Referencia	RF6, RF7, RF8, RF9

Tabla 2.9. Caso de uso Gestionar colección.

CU-7	Gestionar colección
Actor	Administrador
Descripción	El administrador una vez autenticado, puede crear, mostrar, modificar y eliminar colecciones que son empleadas para realizar las peticiones a los DPs.
Referencia	RF10 RF11, RF12, RF13

Tabla 2.10. Caso de uso Indexar documento manualmente.

CU-8	Indexar documento manualmente
Actor	Administrador
Descripción	El administrador indexa los documentos recibidos por el SP,

	guardándolos como nodos en el sistema.
Referencia	RF14

Tabla 2.11. Caso de uso Indexar documento automáticamente.

CU-9	Indexar documento automáticamente
Actor	Reloj
Descripción	El sistema mediante una tarea cron que se ejecuta cada determinado tiempo, indexa los documentos recibidos por el SP, agregándolos como nodos en el sistema.
Referencia	RF15

Tabla 2.12. Caso de uso Indexar documento automáticamente.

CU-10	Actualizar token
Actor	Administrador, Reloj
Descripción	Una vez culminado el proceso de indexación, el campo de fecha de actualización que representa los contenidos colectados en forma de paginado de las fuentes y colecciones de las que se hayan indexado documentos, se modifica con el parámetro <i>resumptionToken</i> devuelto por el DP en el proceso de indexado.
Referencia	RF16

Tabla 2.13. Caso de uso Indexar documento desde archivo.

CU-11	Indexar documento desde archivo
Actor	Administrador
Descripción	El administrador indexa los documentos mediante un fichero XML.
Referencia	RF17

2.6.2 Diagrama de casos de uso del sistema

El diagrama de casos de uso ayuda a comprender gráficamente los procesos del sistema y su interacción con los actores:

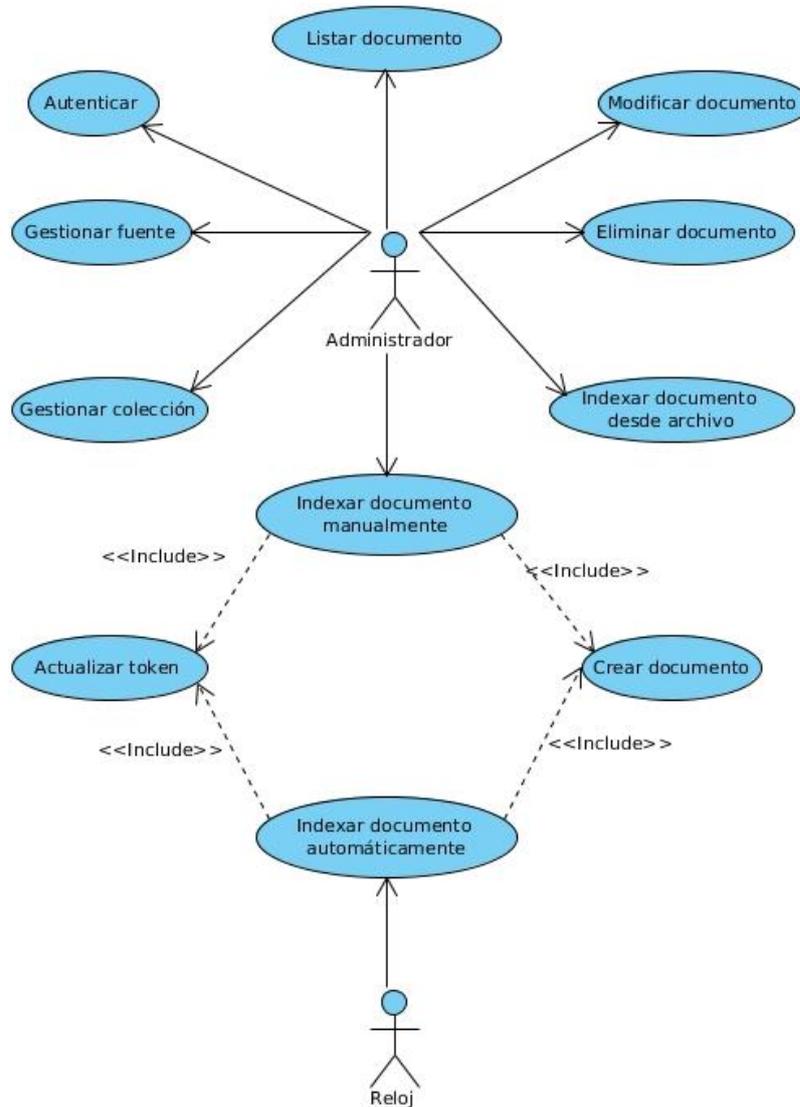


Fig. 2.2: Diagrama de casos de uso del sistema.

2.6.3 Patrones de casos de uso

Estos patrones son utilizados generalmente como plantillas que describen como deberían ser estructurados y organizados los casos de uso. Son patrones que capturan mejores prácticas para modelar casos de uso. Su uso está dado para aumentar la productividad, reutilizar elementos

existentes, evitar el re-trabajo por errores, no invertir tiempo en resolver problemas ya resueltos, aplicar la teoría al trabajo práctico y habilitar las herramientas de soporte para modelar el desarrollo.

Se emplea el patrón CRUD acrónimo del inglés de *Create, Read, Update, Delete*, que consiste en un caso de uso para administrar la información y permite modelar las diferentes operaciones para administrar una entidad, tales como crear, mostrar, modificar y eliminar.

2.6.4 Descripción de los casos de uso arquitectónicamente significativos del sistema

Se describen los casos de usos más significativos del sistema. El resto de las descripciones se encuentran en el [Anexo 1](#).

Tabla 2.14. Descripción del caso de uso *Indexar documento manualmente*.

Objetivo	Crear nodos de Drupal representando a los documentos provenientes del DP.	
Actores	Administrador (Inicia): Indexa documento.	
Resumen	Se envía una petición al DP y a partir de las respuestas en formato XML, se indexa uno o varios documentos.	
Complejidad	Alta.	
Prioridad	Crítico.	
Precondiciones	El administrador tiene que estar autenticado y tiene que existir al menos una colección y/o fuente creada.	
Postcondiciones	Quedó creado un documento.	
Flujo de eventos		
Flujo básico		
	Actor	Sistema
1.	Selecciona la opción de indexar documento.	Muestra las operaciones que se pueden realizar.
2.	Selecciona una de las opciones.	Permite según la opción: <ul style="list-style-type: none"> - Indexar por fuente. Ver Sección 1: "Indexar por

		fuelle”.
		- Indexar por colección. Ver Sección 2: “Indexar por colección”.
3.	Marca la o las colecciones de las que se quiere indexar contenido y presiona el botón de indexar.	Verifica que exista al menos una colección seleccionada.
Sección 2: “Indexar por fuente”		
Flujo básico		
	Actor	Sistema
1.		Lista las fuentes almacenadas en el sistema. Ver Sección 2 del CU-7 Gestionar fuente.
2.	Presiona la opción de “Indexar documento” de una fuente.	Envía peticiones confeccionadas a partir de la url base y el estándar de metadatos que contiene la fuente.
3.		Interpreta los resultados obtenido a partir de la url base.
4.		Indexa los contenidos y notifica su indexación.
Sección 2: “Indexar por colección”		
Flujo básico		
	Actor	Sistema
1.	Selecciona una fuente.	Muestra las colecciones pertenecientes a dicha fuente. Ver Sección 2 del CU-8: Gestionar colección.
2.	Marca las colecciones a indexar.	Envía peticiones confeccionadas a

		partir de la url base y el estándar de metadatos que contiene la fuente de la o las colecciones.
3.		Interpreta los resultados obtenido a partir de la url base.
4.		Indexa los contenidos y notifica su indexación.
Flujos alternos		
1a. Colecciones no seleccionadas		
	Actor	Sistema
1.		Muestra un mensaje de error comunicando que hay que seleccionar al menos una colección.
Relaciones	CU Incluidos	<ul style="list-style-type: none"> - Modificar la o las colecciones y/o fuentes seleccionadas. Ver CU-11: Actualizar token. - Se crea uno o varios documentos. Ver CU-3: Crear documento.
Requisitos no funcionales	Interfaz, seguridad, estándares aplicables.	
Asuntos pendientes	_____	

Tabla 2.15. Descripción del caso de uso Indexar documento automáticamente.

Objetivo	Crear nodos de Drupal representando a los documentos provenientes del DP.
Actores	Reloj (Inicia): Envía una petición cada un tiempo estimado e indexa documentos.

Resumen	Se envía una petición al DP y a partir de las respuestas en formato XML, se indexa uno o varios documentos.	
Complejidad	Alta.	
Prioridad	Crítico.	
Precondiciones	_____	
Postcondiciones	Quedó creado un documento.	
Flujo de eventos		
Flujo básico		
	Actor	Sistema
1.	Chequea los tiempos de actualización de las colecciones y la fecha de última indexación y acciona el indexado de contenido solamente para las colecciones y fuentes que tengan que actualizarse.	Envía peticiones confeccionadas a partir de la url base y el estándar de metadatos que contiene la o las colecciones y fuentes.
2.		Interpreta los resultados obtenido a partir de la url base.
3.		Indexa los contenidos.
Relaciones	CU Incluidos	<ul style="list-style-type: none"> - Modificar la o las colecciones y/o fuentes seleccionadas. Ver CU-11: Actualizar token. - Se crea uno o varios documentos. Ver CU-3: Crear documento.
Requisitos no funcionales	Interfaz, seguridad, estándares aplicables.	
Asuntos pendientes	_____	

Capítulo 3

CONSTRUCCIÓN DE LA PROPUESTA DE SOLUCIÓN Y VALIDACIÓN DE LOS RESULTADOS DEL PROVEEDOR DE SERVICIOS OAI PARA EL SISTEMA DE GESTIÓN DE CONTENIDOS DRUPAL

En el presente capítulo se describen los elementos fundamentales del análisis y diseño de la construcción de la propuesta de solución. Se modelan los diagramas de clases de análisis que participan en la realización de casos de uso. El diseño define los objetos que serán usados en la implementación del sistema y esto es modelado mediante los diagramas de despliegue y de componentes. También se elaboran los casos de pruebas funcionales y se validan los datos.

3.1 Análisis

Este flujo garantiza la refinación y estructuración de los requisitos, profundizándose en el dominio de la aplicación. Esto permite una mayor comprensión del problema para modelar la solución y ofrece un mayor poder expresivo y una mayor formalización. El análisis consiste en obtener una visión del sistema funcional.

3.1.1 Diagrama de clases del análisis

Están reflejados solamente los diagramas de clases del análisis más significativos. Los restantes pueden ser vistos en el [Anexo 2](#).

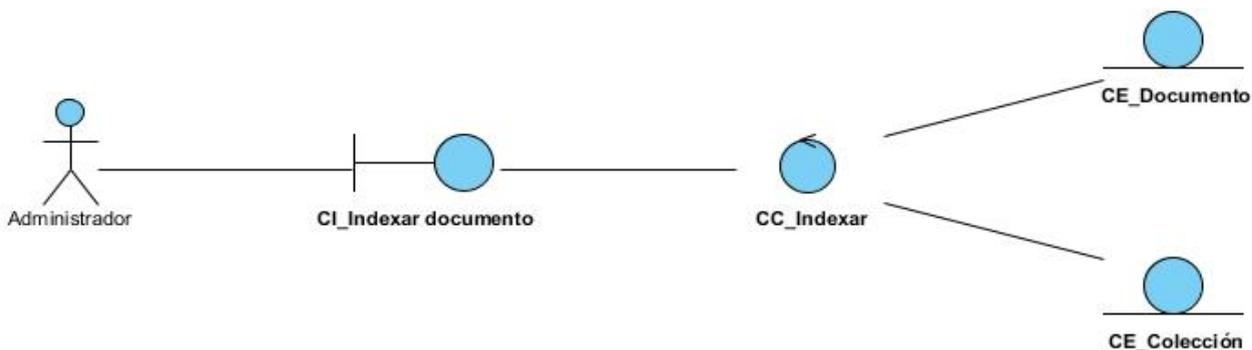


Fig. 3.1: Diagrama de clases del análisis: CU-9 Indexar documento manualmente.



Fig. 3.2: Diagrama de clases del análisis: CU-12 Indexar documento desde archivo.

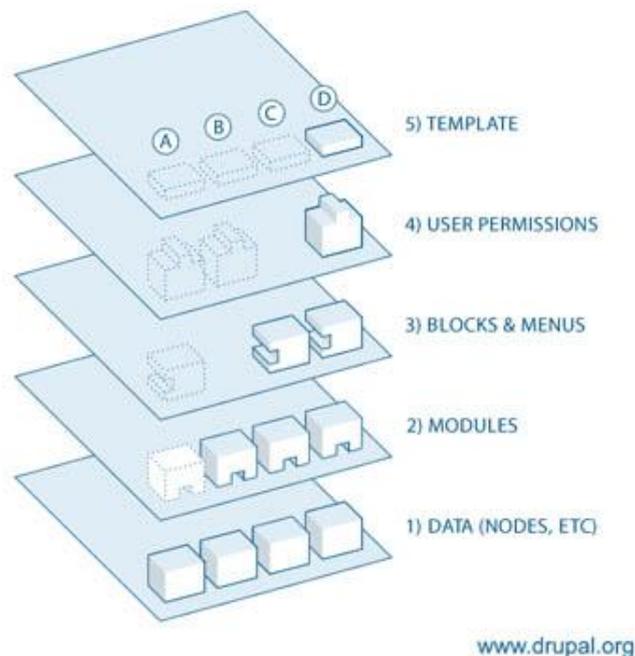
3.2 Diseño

El diseño es un modelo de objetos que describe la realización de los casos de uso, formula los modelos que se centran en los requisitos no funcionales y en el dominio de la solución (Pressman, 2003). Se comprende mejor las relaciones existentes entre los requisitos no funcionales y las restricciones relacionadas con los lenguajes de programación, componentes reutilizables, sistemas operativos, tecnologías de interfaz de usuario, etc. El modelo de diseño es utilizado como entrada en las actividades relacionadas a la implementación y representa a los casos de uso en el dominio de la solución.

3.2.1 Arquitectura de software

Para la implementación de la versión 2.0 del módulo del proveedor de servicios (SP del inglés *service provider*) Iniciativa de Archivos Abiertos - Protocolo para la Transmisión de Metadatos (OAI-PMH del inglés *Open Archives Initiative – Protocol for Metadata Harvesting*) para el Sistema de Gestión de Contenidos (CMS del inglés *Content Management Systems*) Drupal se utiliza una arquitectura en capas como se muestra en la figura 3.3 y descrita según lo tomado por Rawtani y Chidambaram (2009):

La primera capa representa la información almacenada en los nodos (fuentes, colecciones y documentos). En la segunda se encuentra ubicado el módulo “*service_provider*” y los restantes módulos que le otorgan el correcto funcionamiento al SP. La tercera está descrita por los ganchos de menús (*hook menu*) trabajados en la implementación que reciben y gestionan todas las peticiones efectuadas y por el bloque de administración para el SP. La cuarta recoge los permisos de usuarios creados en el gancho de privilegios (*hook perm*) y la quinta y última capa maneja la presentación visual de los datos administrados por los ficheros del tema y las vistas.



Fuente: <http://www.drupal.org>. [Artículo en línea, 1 de Mayo de 2012].

Fig. 3.3: Arquitectura de software en capas.

3.2.2 Diagrama de clases

A continuación se muestra el diagrama de paquetes que genera el CMS Drupal. A partir del mismo se representan los diagramas de clases de diseño según el sistema que se describe. Las clases que serán descritas son solamente aquellas que se relacionan directamente con el diseño del sistema que se está modelando.

- Includes: contiene ficheros con las funciones necesarias para el funcionamiento de Drupal, por ejemplo *Database*, que provee funcionalidades de acceso a la base de datos tanto para MySQL como PostgreSQL.
- Themes: en esta carpeta se hallan las distintas plantillas de Drupal, pueden incluirse también en `sites/all/themes`.
- Modules: este directorio contiene los módulos que hacen posible el funcionamiento del CMS, pueden incluirse además en la carpeta `sites/all/modules`.
- Scripts: posee un grupo de ficheros imprescindibles para la correcta visualización de los sitios construidos con Drupal.

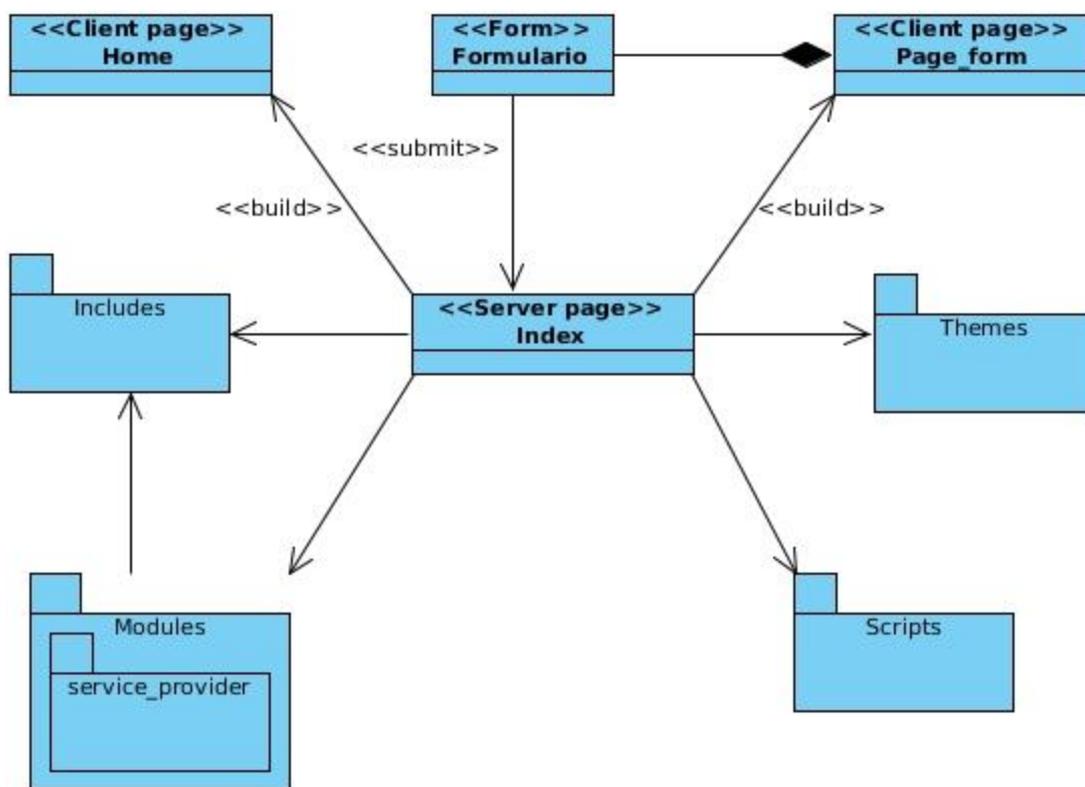


Fig. 3.4: Diagrama de clases del diseño por paquetes.

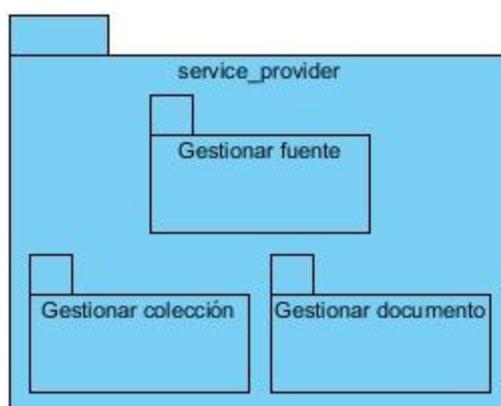


Fig. 3.5: Subpaquete service_provider.

3.2.3 Diagrama de clases del diseño

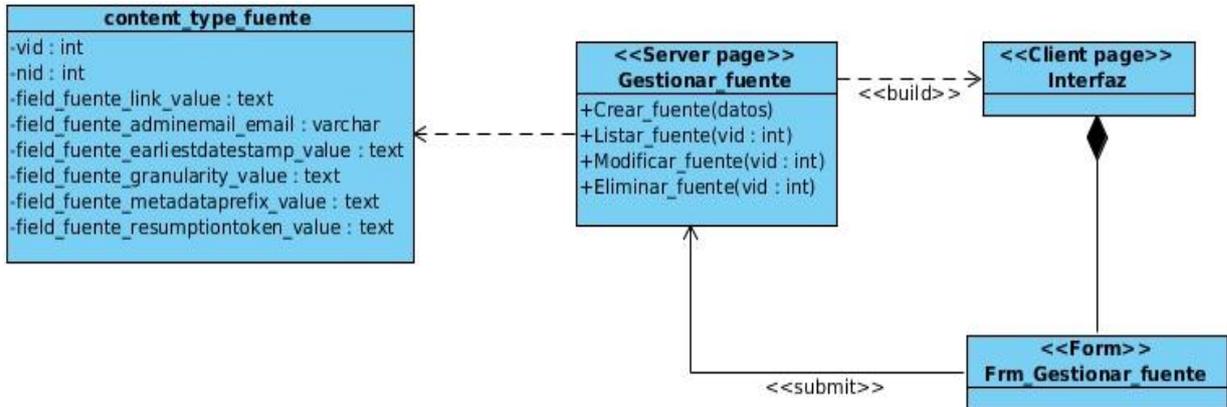


Fig. 3.6: Diagrama de clases del diseño. Subpaquete Gestionar fuente.

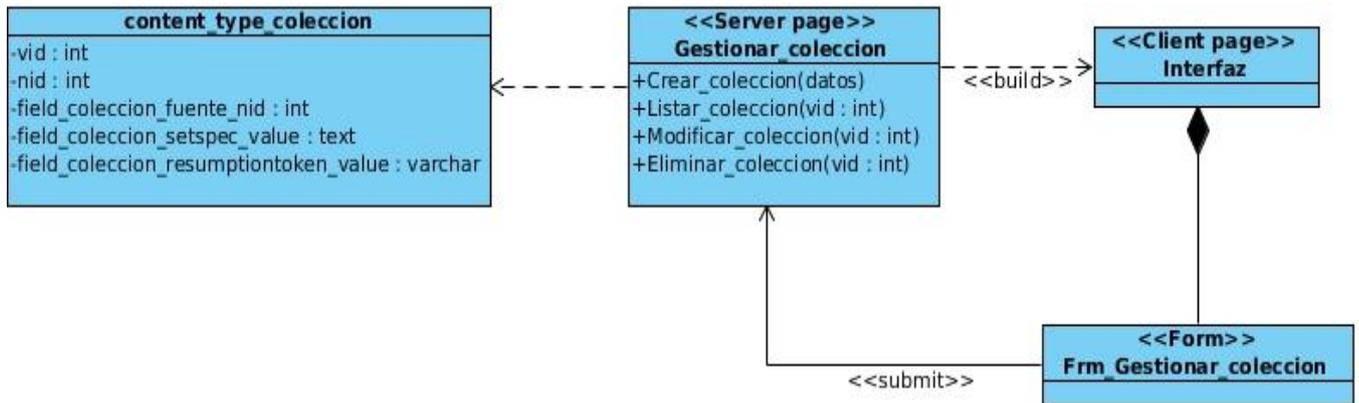


Fig. 3.7: Diagrama de clases del diseño. Subpaquete Gestionar coleccion.

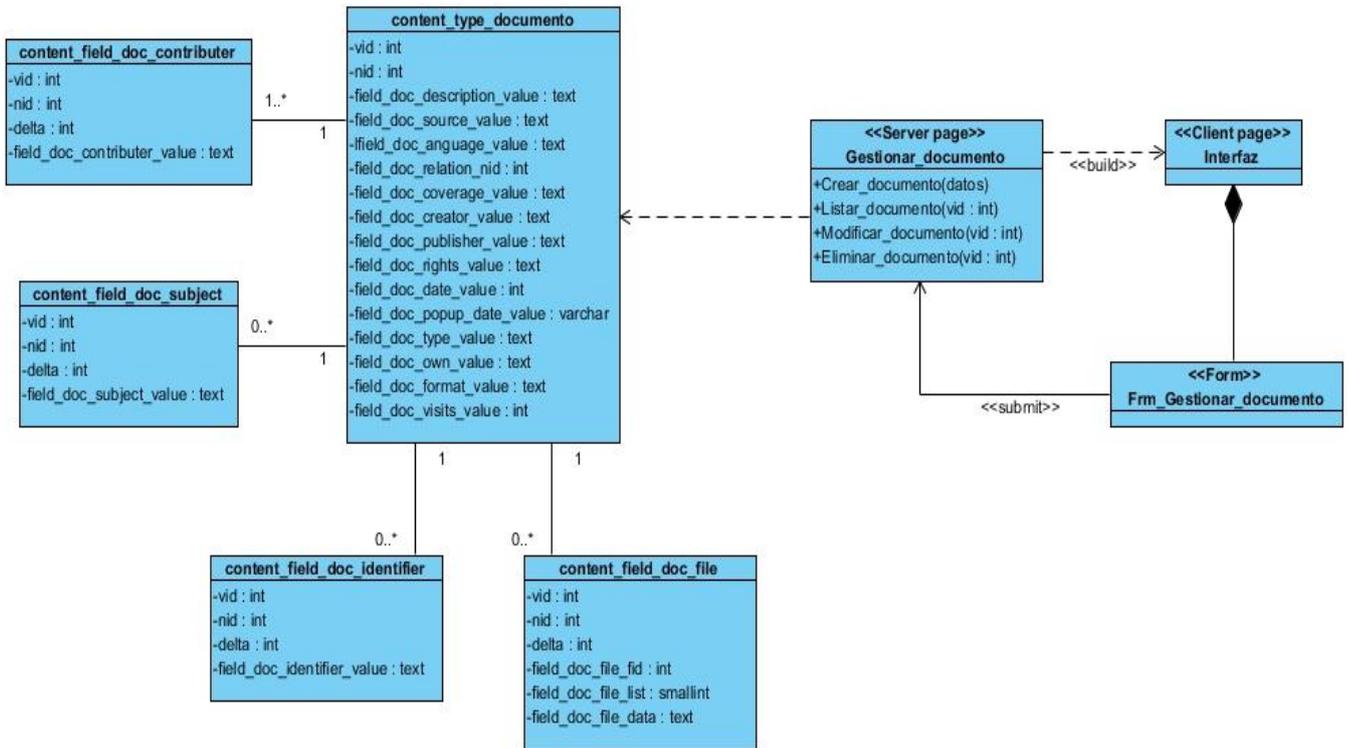


Fig. 3.8: Diagrama de clases del diseño. Subpaquete Gestionar documento.

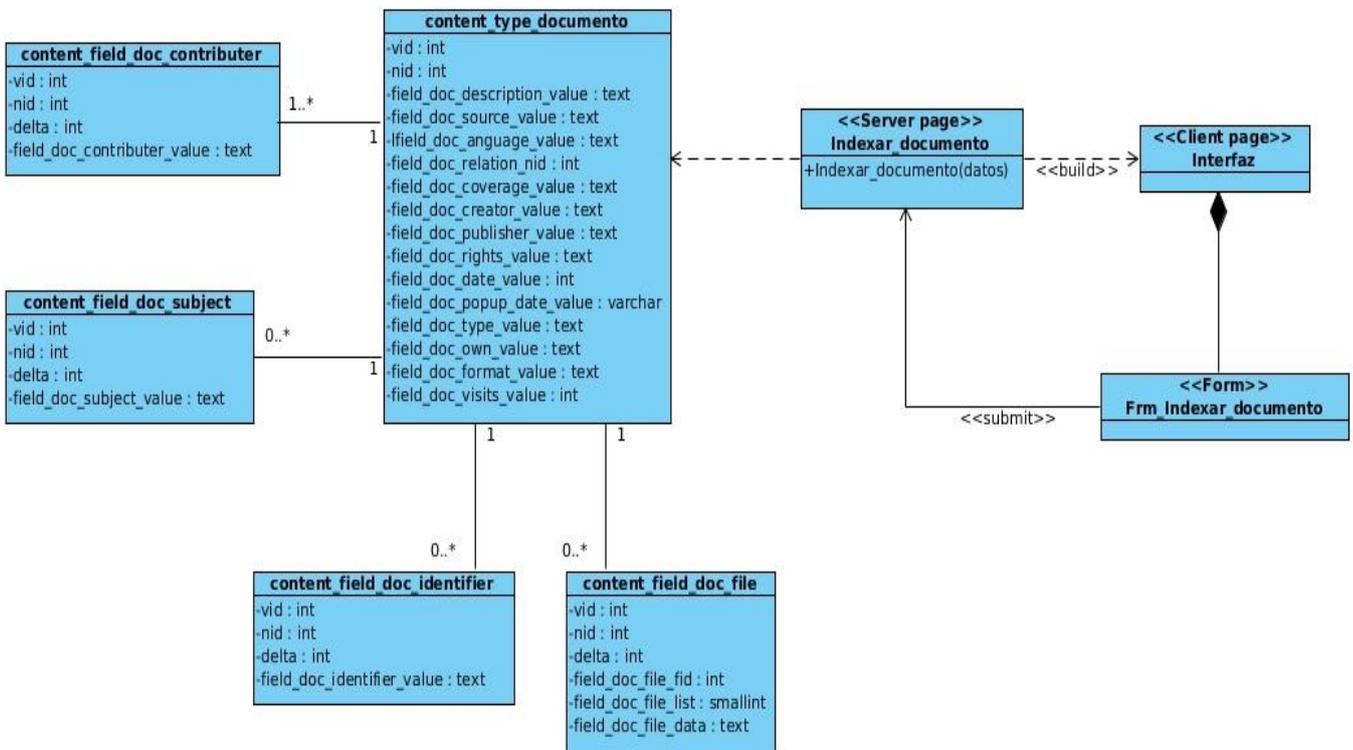


Fig. 3.9: Diagrama de clases del diseño. Subpaquete Indexar documento.

3.3 Diseño de la base de datos

El principal objetivo del diseño de la base de datos es generar las tablas que modelan los registros en los que se guarda la información del sistema. A continuación se muestra el diseño de la base de datos del sistema propuesto a través de los diagramas de clases persistente y el esquema de la base de datos generados a partir de este, con el modelo de datos.

3.3.1 Diagrama de clases persistentes

El diagrama de clases persistentes muestra todas las clases capaces de mantener su valor en el espacio y en el tiempo. Las clases persistentes, sus atributos y sus relaciones son representados a través del siguiente diagrama.

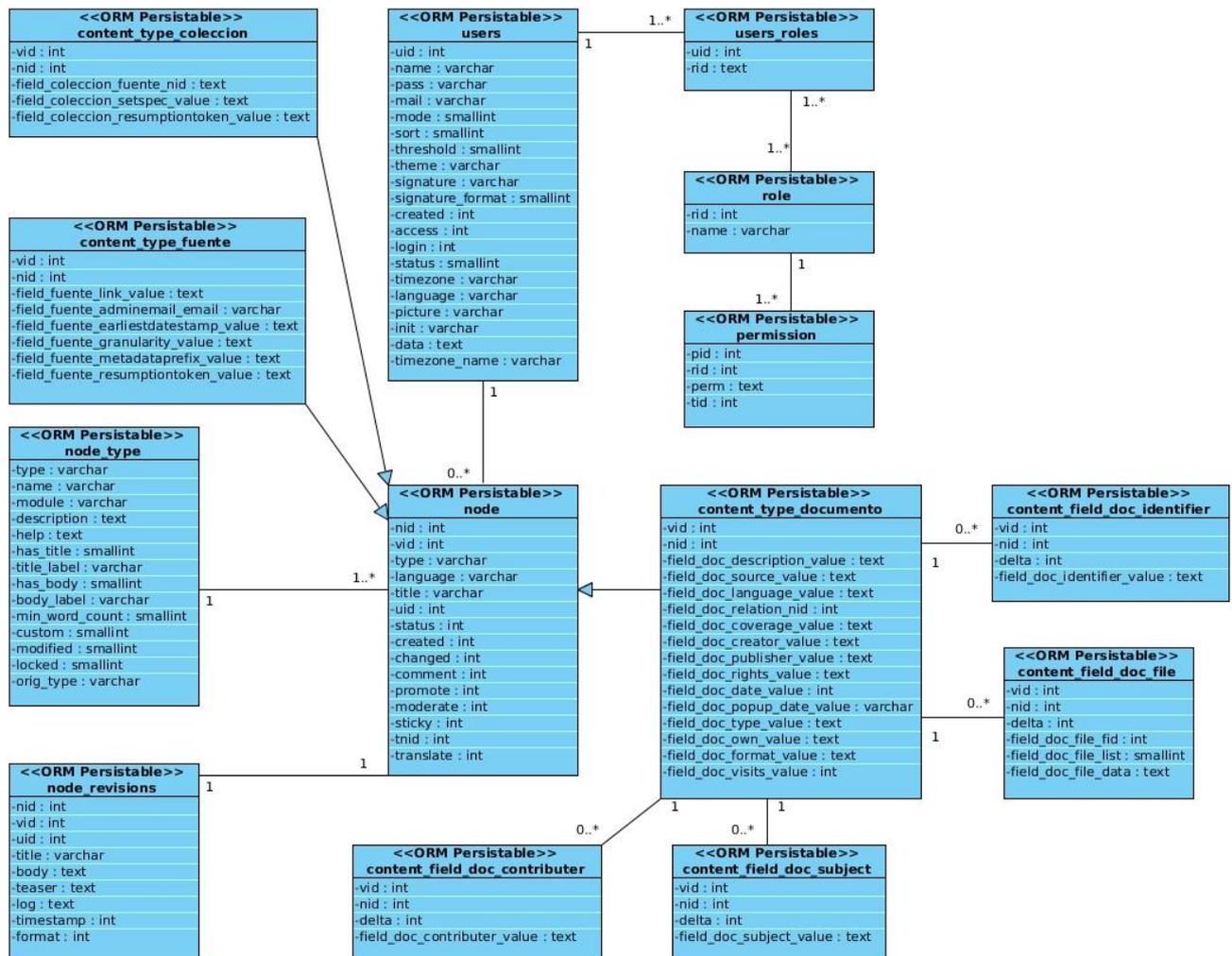


Fig. 3.10: Diagrama de clases persistentes.

3.3.2 Modelo de datos

El modelo de datos muestra la estructura física de las tablas de la base de datos, obtenido a partir del diagrama de clases persistentes. En el mismo sólo se representan las nuevas entidades que fueron usadas para dar solución al problema que se modela.

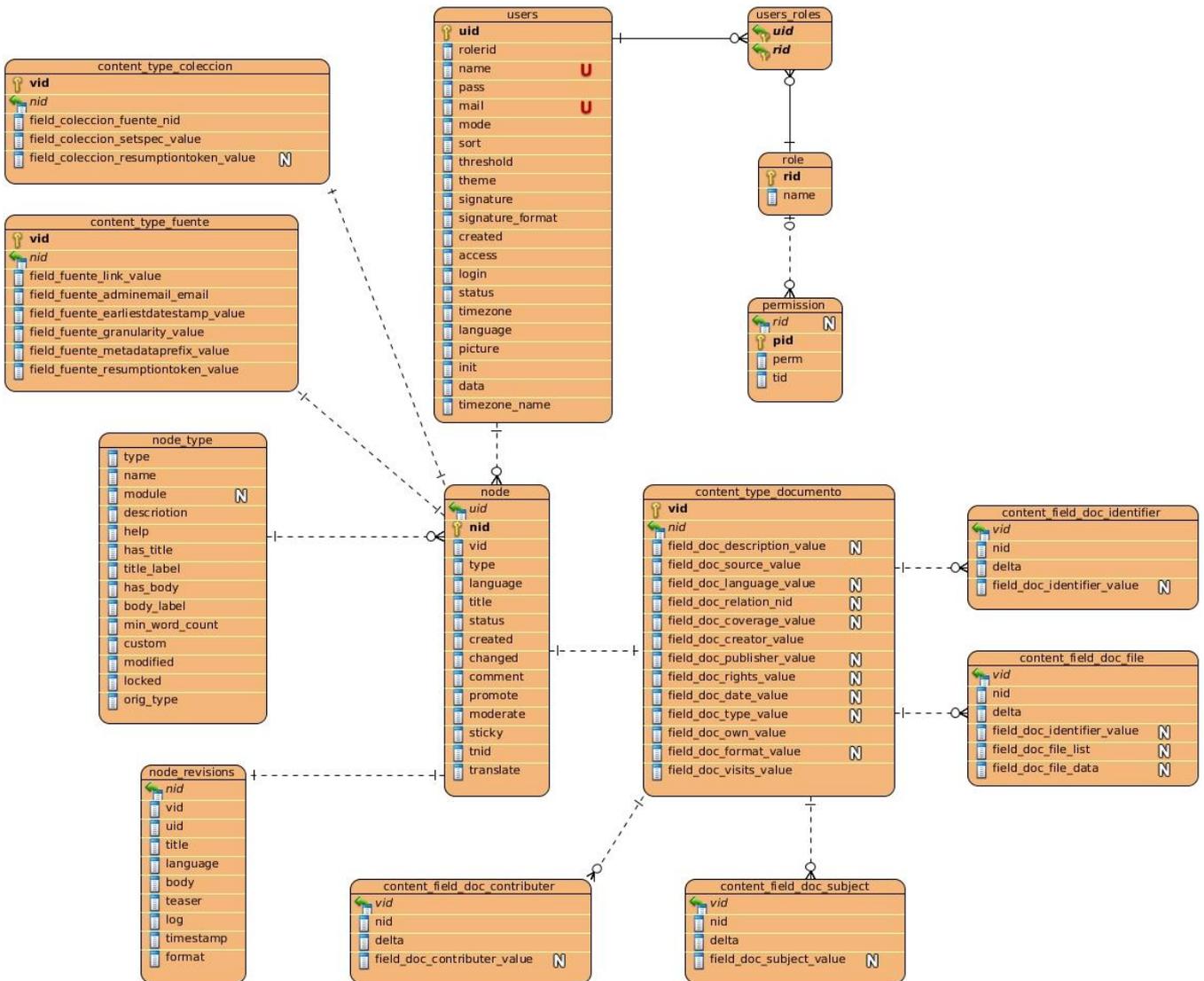


Fig. 3.11: Modelo de datos.

La descripción de las tablas de los tipos de contenidos que se emplearon para el desarrollo de la aplicación puede encontrarse en el [Anexo 3](#).

3.4 Modelo de implementación

Esta etapa detalla el sistema en términos de componentes, ficheros de código fuente y binario, ejecutable y scripts dentro del flujo de implementación que propone el Proceso Unificado de Rational (RUP del inglés *Rational Unified Process*). Se define la organización del sistema en términos de subsistemas de implementación organizados en capas. El modelo de implementación describe cómo los elementos del modelo de diseño se implementan en términos de componentes y está conformado por los diagramas de componentes y despliegue; estos describen los componentes a construir y dependencias entre nodos físicos en los que funcionará cualquier sistema que utilice el módulo a desarrollar.

3.4.1 Diagrama de despliegue

Un diagrama de despliegue modela la arquitectura en tiempo de ejecución de un sistema, muestra la configuración de los elementos de hardware (nodos) y muestra cómo los elementos y artefactos del sistema se visualizan en esos nodos. El objetivo fundamental de este diagrama es llevar la aplicación a los componentes que integran la parte material de una computadora para hacer funcional dicha aplicación. Es un despliegue de los recursos físicos que se necesitarán para que el sistema funcione correctamente.

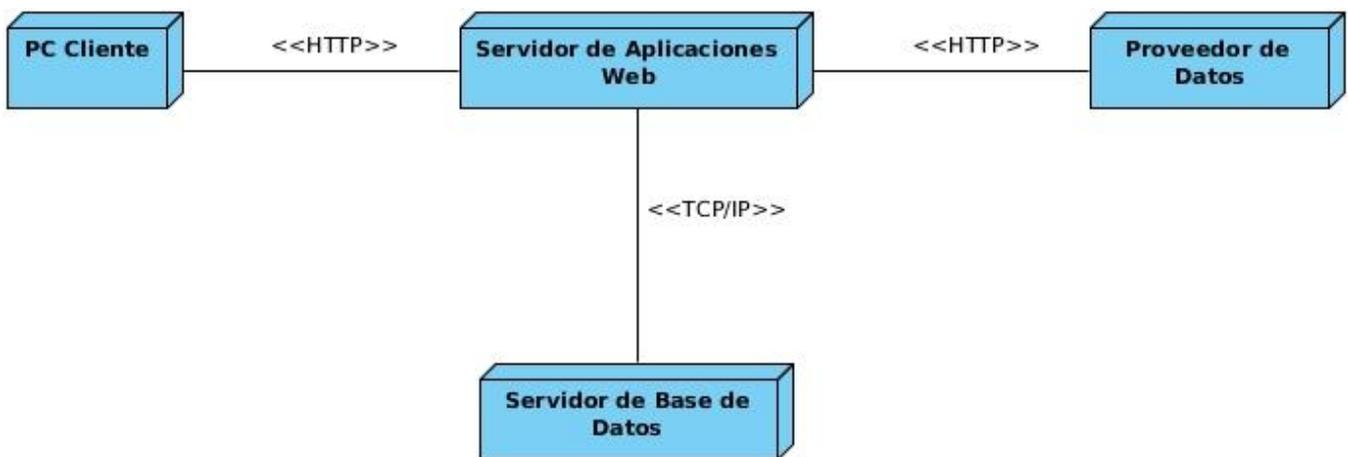


Fig. 3.12: Diagrama de despliegue.

A continuación se describen cada uno de los nodos presentes en el diagrama anterior:

PC Cliente: representa la computadora personal del cliente que se conecta al Servidor de Aplicaciones Web. La misma se comunica al servidor mediante el Protocolo de Transferencia de Hipertexto (HTTP del inglés *Hypertext Transfer Protocol*).

Servidor de Aplicaciones Web: representa el servidor sobre el cual se encuentra corriendo la aplicación Web. Este accede al Servidor de Base de Datos para el manejo de información.

Servidor de Base de Datos: es la representación de la base de datos perteneciente a la aplicación.

Proveedor de Datos: personifica al sistema encargado de exponer los metadatos a partir de las peticiones generadas desde el Servidor de Aplicaciones Web a través del protocolo HTTP.

3.4.2 Diagrama de componentes

Los diagramas de componentes describen los elementos físicos del sistema y sus relaciones, representan todos los elementos que entran en la fabricación de aplicaciones informáticas; son usados para estructurar el modelo de implementación en términos de subsistemas de implementación y mostrar la relación entre ellos. Es un grafo de componentes unidos a través de relaciones que pueden ser de ejecución o compilación. La figura 3.13 muestra el diagrama de componentes que genera Drupal para estructurar los subsistemas de implementación.

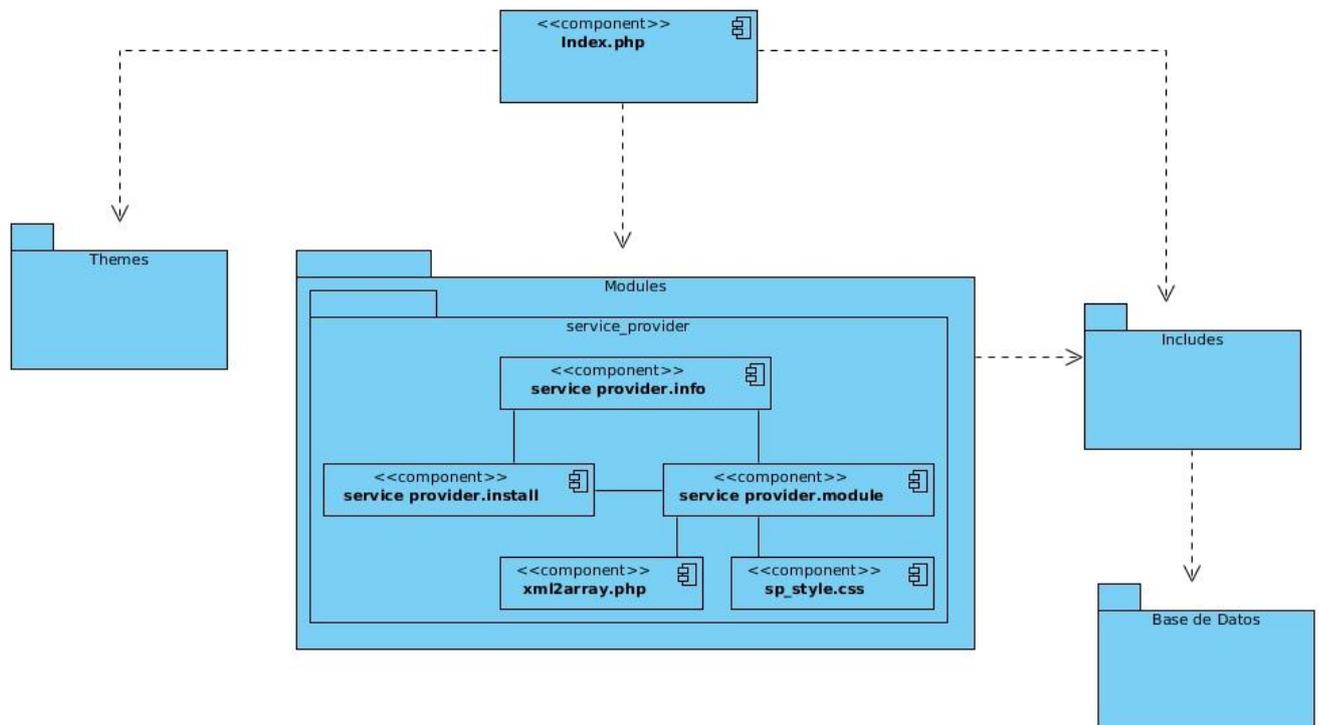


Fig. 3.13: Diagrama de componentes.

3.4.2.1 Descripción de los componentes

Tabla 3.8. Descripción de los componentes.

Componente	Descripción
Index.php	Página PHP que sirve la arquitectura de Drupal. Su contenido responde en cada caso a las peticiones que se estén realizando.
Themes	Conjunto de temas que representan la forma en que se visualiza el sistema.
Modules	Conjunto de módulos que le dan la funcionalidad a Drupal.
service_provider	Archivo donde se encuentran los ficheros que componen el módulo.
service_provider.info	Fichero que almacena la información principal del módulo.
service_provider.install	Fichero que alberga datos que se insertan en el sistema como la

	estructura de los contenidos una vez que se instala el módulo.
service_provider.module	Fichero que contiene el código del módulo implementado.
xml2array	Librería que interpreta ficheros XML.
sp_style	Fichero que maneja los estilos del resultado de la implementación del módulo.
Includes	Conjunto de funciones, que permiten la conexión al Servidor Web y a la Base de Datos.
Base de Datos	Representa la base de datos, en este caso PostgreSQL

3.5 Diferencias entre las versiones de los módulos implementados

La siguiente tabla muestra las diferencias existentes entre la implementación de ambas versiones del módulo del SP. En ella se evidencia la superioridad en desarrollo, estructura y funcionamiento de la segunda versión.

Tabla 3.9. Diferencias entre versiones de módulos.

Versión 1.0	Versión 2.0
Para su instalación, el módulo depende de que estén creados los tipos de contenidos.	Las dependencias se crean en el proceso de instalación.
Eficiencia de los algoritmos.	Mejora en la eficiencia de los algoritmos, producto de la introducción del uso de nuevos verbos y control de los contenidos duplicados.
Presenta interfaz de usuario.	Presenta interfaz de usuario mejorada.
Indexado solamente manual.	Indexado manual y automático.
Sólo se almacena el nombre y la url base de las fuentes.	Se almacena mayor cantidad de datos de las fuentes para su uso posterior.
No se verifican los formatos de metadatos de las fuentes.	Se comprueban los formatos de metadatos con el uso del <i>ListMetadataFormats</i> de las fuentes a

	recolectar.
Indexación basada en <i>ListRecords</i> para almacenar documentos.	Se incluye el tratamiento de colecciones mediante el uso del verbo <i>ListSet</i> .
Continuidad de la recolección por fechas.	Se incorpora el uso del <i>resumptionToken</i> para la continuidad del indexado.

3.6 Pruebas

En la elaboración de una aplicación Web, el proceso de prueba es imprescindible en la detección de errores o fallas. Las pruebas juegan un papel fundamental en la garantía del producto. Una selección cuidadosa de los datos de prueba, puede ofrecer una elevada confianza en cuanto al desempeño que posee el programa. Esto, junto a un mecanismo de comprobación de errores, tiende a convertir un producto en más confiable. Estas se realizan con la finalidad de obtener un conjunto de pruebas que brinden la mayor cantidad de información posible para poder encontrarle defectos al sistema.

3.6.1 Pruebas internas

Para aplicarle las pruebas al sistema, se usó el método de prueba de caja negra (prueba de comportamiento): son diseñadas para validar los requisitos funcionales sin fijarse en el funcionamiento externo del programa.

Se realizaron los diseños de casos de prueba basado en casos de uso para determinar errores en la aplicación. Se validaron las funcionalidades introduciendo distintos tipos de datos, válidos y no válidos para comprobar cuan viable es la misma. Ver resultado de las pruebas en el [Anexo 4](#).

Al sistema se le realizaron tres iteraciones de prueba. En la primera fueron identificadas diez no conformidades; todas resueltas. Se ejecutó una segunda iteración donde se detectaron cinco no conformidades las cuales fueron resueltas satisfactoriamente. Finalmente se realizó una tercera y última iteración de pruebas la cual arrojó tres no conformidades y todas fueron resueltas. Ver no conformidades en el [Anexo 5](#).

3.6.2 Tiempo de recolección

Con el objetivo de demostrar que el proceso de indexación del módulo en su versión 2.0 funciona más rápido que la versión anterior se midió el tiempo de indexación de documentos con ambos módulos por separados, usando el navegador Web Mozilla Firefox 11.0, el Sistema de Gestión de Base de Datos (SGBD) PostgreSQL 9.1.3 y un mismo ordenador. Esto queda demostrado a partir de la tabla 3.9 y la figura 3.14. En ellas queda reflejado que para el módulo versión 2.0, a medida que aumentan los documentos a almacenar, el tiempo de recolección es mucho más rápido que para la versión 1.0.

Tabla 3.9. Tiempo de demora de indexación de documentos.

Documentos	100	300	600	900	1000	2000	3000	4000	5000
Tiempo en segundos para Versión 1.0	32	93	187	285	319	675	1072	1508	1985
Tiempo en segundos para Versión 2.0	9	28	56	83	93	182	266	347	423

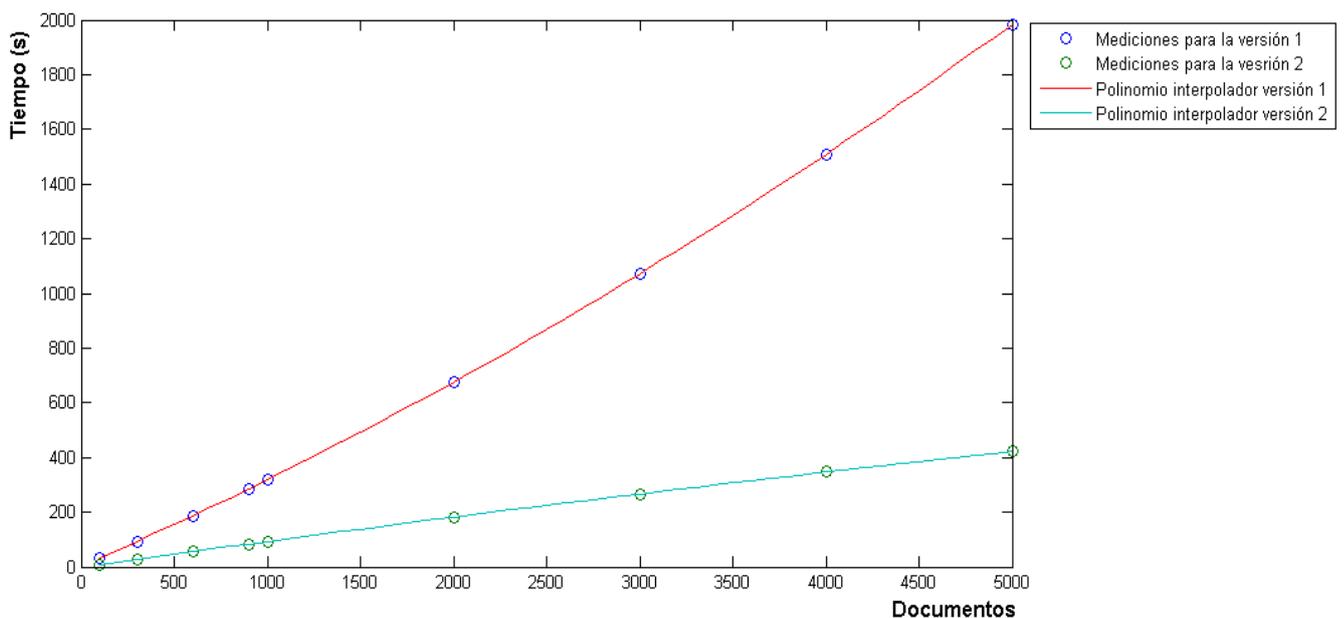


Fig. 3.14: Comportamiento del tiempo de indexado mediante una función interpoladora.

CONCLUSIONES

- Para la distribución de contenidos a través de las redes informáticas han surgido protocolos para el intercambio de información como el OAI-PMH, muy difundido y usado para estos propósitos, que está dividido en dos roles: proveedores de datos y de servicios, garantizando un acceso libre y completo a la información.
- El CMS Drupal constituye una herramienta muy potente en el desarrollo de aplicaciones Web y posee altas prestaciones para la correcta implementación de un proveedor de servicios.
- La arquitectura seguida en la implementación permite incorporar futuras nuevas funcionalidades.
- La implementación del proveedor de servicios OAI en el CMS Drupal acelera el proceso de búsqueda de información de carácter científico de forma rápida y de libre de acceso para los usuarios.
- El desarrollo del módulo en su segunda versión recupera contenidos de forma más rápida y con mayor nivel de organización.

RECOMENDACIONES

- Emplear el módulo desarrollado en bibliotecas, repositorios o revistas digitales que estén creadas bajo el CMS Drupal en su versión 6.x.
- Implementar el módulo “service_provider” para la versión 7.x de Drupal.
- Continuar con el estudio de las características y funcionalidades del protocolo OAI-PMH para aprovechar al máximo las potencialidades que este ofrece.
- Desarrollar un mecanismo de optimización que permita la recuperación de los documentos en copia digital para su consulta directa.

REFERENCIAS BIBLIOGRÁFICAS

Abián, M. 2006. El futuro de la Web, XML, RDF, ontologías y la Web semántica. [En línea] [Citado el: 7 de Diciembre de 2011.] Disponible en: <http://www.hosteltur.com/web/uploads/f3dfc92eb1db041e.pdf>.

ANSI/NISO. 2003, Information Retrieval (Z39.50): Application Service Definition and Protocol Specification. ANSI/NISO Z39.50-2003. EEUU: ANSI. ISSN: 1041-5653.

Apache. 2010. Una Introducción a APACHE. [En línea]. Ciberaula, 2010. [Citado el: 12 de Enero de 2012.] Disponible en: http://linux.ciberaula.com/articulo/linux_apache_intro.

Armas, A. R.; Chamorro, G. A. y Montes, B. M. 2007. Desde ISO 9001 hacia CMMI, pasos para la mejora de los procesos y métricas. RPM-AEMES, Vol. 4. p. 18-25.

Barrueco, J. M. 2003. SUBIRATS COLL, I. OAI-PMH: protocolo para la transmisión de contenidos en internet. El profesional de la información, 2003, marzo-abril, v. 12, n. 2, p. 104.

Barrueco, J. M. 2006. Open Archives Initiative Protocol for Metadata Harvesting (OAI-PMH): descripción, funciones y aplicación de un protocolo. 2006. [En línea] [Citado el: 29 de Noviembre de 2011.] Disponible en: <http://eprints.rclis.org/bitstream/10760/4093/1/cardedeu.pdf>.

BNC. 2009. Manual para la creación de metadatos usando Dublin Core. Biblioteca Nacional de Chile. Com. De Met. De la Bib. Nac. de Chile. Santiago. [En línea] [Citado el: 29 de Noviembre de 2011.] Disponible en: http://www.aatespanol.cl/taa/publico/ftp/archivo/MANUAL_WEB.pdf.

Bueno, de la F. R y Rodríguez, M. D. La Iniciativa de Archivos Abiertos (OAI): situación y perspectivas en España y Latinoamérica. [En línea]. Bogotá: Rojas Eberhard, 2007. [Citado el: 2 de Diciembre de 2011.] Capítulo 8. Herramientas de software para OAI-PMH. Disponible en: <http://e-archivo.uc3m.es/bitstream/10016/9088/1/CAPITULO8-OAI-Software.pdf>. ISBN 978-958-9121-89-4.

Calderín, C. M. 2007. SISTEMA DE INFORMACIÓN DOCUMENTAL EN LA ACADEMIA: SITUACIÓN ACTUAL Y PERSPECTIVAS FUTURAS, CASO UCAB. [En línea]. Universidad Católica Andrés Bello. [Citado el 24 de Noviembre de 2011.] Disponible en: www.cienciared.com.ar/ra/usr/3/333/n6_v1pp51_76.pdf.

Chalon, P. 2008. Drop in: Drupal for libraries. Journal of the European Association for Health Information and Libraries (JEAHIL), 2008, vol. 4, no. 3, pp. 40-41. [En línea] [Citado el: 3 de Diciembre de 2011.] Disponible en: <http://eprints.rclis.org/handle/10760/12208>.

De Giusty, M. 2006. Tecnologías para propagar los contenidos de una Biblioteca Digital. En Málaga. 2006. [En línea] [Citado el: 1 de Diciembre de 2011.] Disponible en: www.uma.es/ficha.php?id=79799.

Diccionario Enciclopédico. 2001. Editorial, Océano Grupo. Diccionario Enciclopédico. Barcelona : Milenio, 2001. 1045 pp.

Donoso y Vergara, 2011. Alexandria. [En línea]. Companion Corporation, Inc, 2011. [Citado el: 2 de Diciembre de 2011.] Disponible en: <http://www.infoconsultores.com.mx/Automatizacion/Alexandria.htm>.

Echebarría, G. 2007. OAI-PMH, protocolo para la transmisión de metadatos. [En línea] [Citado el: 1 de Diciembre de 2011.] Disponible en: www.slidefinder.net/.../OAI_PMH_Protocolo_transmisi3n_metadatos/OAI-PMH.../8058030.

Egiguren, R. 2010. Introducción a Drupal. [En línea]. Slide Share Inc., 2011 [Citado el: 3 de Diciembre de 2011.] Disponible en: <http://www.slideshare.net/keopx/introduccion-a-drupal-2971232>.

Euguíluz, P. J. 2009. Introducción a Ajax. [En línea] [Citado el: 15 de Mayo de 2012.] Disponible en: <http://www.librosweb.es>.

Fernández, F. M. M. y Abreu, B. Y. 2010. Biblioteca digital con el uso del sistema de gestión de contenidos Drupal, 2010. [En línea] [Citado el: 17 de Octubre de 2011.] Disponible en: http://tallertematico.fordes.co.cu/index.php/ttca/xst_ca/paper/view/52.

Fernández, F. M. M.; Abreu, B. Y. y Dominguez, C. L. 2011. El protocolo OAI-PMH, componente tecnológico para el acceso abierto. [En línea] [Citado el: 1 de Diciembre de 2011.] Disponible en: <http://www.acimed.sld.cu/index.php/acimed/article/view/150/140>.

Guajardo, A. 2010. Z39.50 y OAI-PMH: Protocolos de Transferencia y Recuperación de Información. En: XV Conferencia Internacional de Bibliotecología. Panorama de las Bibliotecas y la Información en el bicentenario; Centro Cultural Estación Mapocho: 2-4 de noviembre de 2010. Santiago de Chile; 2010. 13p.

Gómez, D. L. 2005. La Iniciativa de Archivos Abiertos (OAI), un nuevo paradigma en la comunicación científica y el intercambio de información. Revista Códice, 2005, vol. 4, nº 25.

Gómez, D. L. 2007. Interoperabilidad en los Sistemas de Información Documental: La Información Debe Fluir. Códice, enero- junio, 2007, v. 3, n. 1, 28 pp.

Howe, D. 2010. Free on-line Dictionary of Computing (FOLDOC). [En línea] [Citado el: 9 de Mayo de 2012.] Disponible en: <http://wombat.doc.ic.ac.uk/foldoc/>.

Lagoze, C. 2005. OAI-PMH. Implementation Guidelines - Guidelines for Harvester Implementers. [En línea] [Citado el: 1 de Diciembre de 2011.] Disponible en: <http://www.openarchives.org/OAI/2.0/guidelines-harvester.htm>.

Lazinger, S. 2001. Digital Preservation and Metadata: History, Theory, Practice. Englewood, Colorado: Libraries Unlimited. 359 pp. ISBN-1-56308-777-4.

López, E.; Cabrera, C. A. M. y Valencia, A., L.E. 2007. Introducción a la Calidad del Software. Pereira : s.n., 2007, Scientia et Technica, Vol. 1. p. 18-25.

Muñoz, C. V. 1999. Gestión y Planificación de Sistemas y Servicios de Información. En: Juan Carlos Suárez, V. J. C y García, G. A. Introducción a la documentación informativa y periodística. Alcalá de Guadaíra, Sevilla : Madrid, S.L, 1999.

Melero, R. 2005. Acceso abierto a las publicaciones científicas: definición, recursos, copyright e impacto. El profesional de la Información, 2005, vol. 14, n 4, p. 255-264. [En línea] [Citado el: 27 de Noviembre de 2011.] Disponible en: <http://www.elprofesionalde lainformacion.com/contenidos/2005/julio/3.pdf>. ISSN 1386-6710.

Méndez, E. y Senso, J. A. 2004. Unidad de Autoformación. SEDIC, Introducción a los metadatos : Estándares y Aplicación. [En línea] [Citado el: 29 de Noviembre de 2011.] Disponible en: <http://www.sedic.es/autoformacion/metadatos/tema1.htm>.

Méndez, Z. y Torréns, R. 2003. Manejo de información básica de datos y metadatos. [En línea] [Citado el: 29 de Noviembre de 2011.] Disponible en: http://www.gwpcentroamerica.org/viejo/gwp/sp/content/tool_802D2218-3FFA-49C8-840C-C2E88842438C.html.

Merino V.; Gaytán F. y Garzón A. 2003. Proceso de mejora continua. [En línea] [Citado el: 28 de Febrero de 2012] Disponible en: <http://www.fundacioncetmo.org/fundacion/publicaciones/transporte.viajeros/procesos.mejora.pdf>.

NetBeans. 2012. NetBeans IDE - The Smarter Way to Code. [En línea]. Oracle Corporation and/or its affiliates, 2012. [Citado el: 8 de Enero de 2012.] Disponible en: <http://netbeans.org/features/index.html>.

NetCraft. 2012. Internet Research, Anti-Phishing and PCI Security Services. [En línea]. Netcraft Ltd 2012. [Citado el: 10 de Mayo de 2012.] Disponible en: <http://news.netcraft.com/>.

Oracle. 2011. MySQL: Why MySQL? [En línea]. Oracle Corporation and/or its affiliates, 2011. [Citado el: 2 de Diciembre de 2011.] Disponible en: <http://www.mysql.com/why-mysql/>.

Pérez, D. 2007. ¿Qué es JavaScript? [En línea]. Maestros del Web, 3 de Julio de 2007. [Citado el: 11 de Enero de 2012.] Disponible en: <http://www.maestrosdelweb.com/editorial/%C2%BFque-es-javascript/>.

Ponjuan, D. G.; Mena, M.; Villardefrancos, M. C.; León, M. y Martí. Y. 2004. Sistemas de Información: principios y aplicaciones. La Habana : Félix Varela, 2004. 138 pp.

Postgresql. 2011. Postgresql: About. [En línea]. PostgreSQL Global Development Group, 1996.[Citado el: 2 de Diciembre de 2011.] Disponible en: <http://www.postgresql.org/>.

Pressman, R. 2003. Ingeniería del Software. Un enfoque práctico. 6ª. ed. 2003. 900 pp. ISBN: 9701054733.

Redondo, H. A. 2011. Adaptación automática de diálogos de agentes conversacionales pedagógicos a los rasgos de la personalidad del estudiante. [En línea]. Universidad Rey Juan Carlos. 2011. [Citado el: 10 de Mayo de 2012] Disponible en: <http://eciencia.urjc.es/handle/10115/5537#preview>.

Rawtani, M. R. y Chidambaram, S. S. 2009. Drupal: The Open Source Content Management System Software Suit For Library With Library 2.0 Features. [En línea]. 21 de Mayo de 2012. [Citado el: 22 de Mayo de 2012.] Disponible en: <http://ir.inflibnet.ac.in/dxml/handle/1944/1021>.

Siza, R. J. P. 2010. Estudio, análisis, evaluación e implementación de un protocolo de intercambio de contenidos sobre internet para la interoperabilidad de repositorios de la Biblioteca Digital en la Universidad Nacional de Colombia con un proveedor de servicios de contenidos. Tesis (Máster en Telecomunicaciones). [En línea]. Bogotá D.C., Colombia : Universidad Nacional de Colombia, Facultad de Ingeniería, 2010. 124 p. [Citado el: 29 de Noviembre de 2011.] Disponible en: <http://www.bdigital.unal.edu.co/1758/1/299759.2010.pdf>.

Tramullas, J. 2010. Drupal para bibliotecas y archivos. [En línea] [Citado el: 6 de Enero de 2012.] Disponible en: <http://eprints.rclis.org/bitstream/10760/14400/1/drupalbibcompleto.pdf>. ISBN: 978-84-613-9611-5.

UNE 50113-1:1992. Asociación Española de Normalización y Certificación. Documentación e información. Vocabulario. Parte 1: conceptos fundamentales. Madrid, España :1992, 18 pp.

UNE-EN ISO 2789:1996. Española de Normalización y Certificación. Información y documentación. Estadísticas internacionales de bibliotecas. (ISO 2789:1991). Madrid, España: 1996, 20pp.

Visual Paradigm. 2011. Visual Paradigm for UML - UML tool for software application development.[En línea]. 12 de Diciembre de 2011. [Citado el: 6 de Enero de 2012.] Disponible en: <http://www.visual-paradigm.com/product/vpuml/>.

W3C. 2008. Guía Breve de XHTML. W3C. [En línea]. 7 de Febrero de 2008. [Citado el: 6 de Enero de 2012.] Disponible en: <http://www.w3c.es/divulgacion/guiasbreves/XHTML>.

W3C. 2005. Guía Breve de CSS. W3C. [En línea].9 de Febrero de 2005. [Citado el: 6 de Enero de 2012.] Disponible en: <http://www.w3c.es/divulgacion/guiasbreves/HojasEstilo>.

w3schools. 2012. PHP Introduction. w3schools. [En línea]. Refsnes Data, 1999. [Citado el: 2 de Diciembre de 2011.] Disponible en: http://w3schools.com/php/php_intro.asp.

BIBLIOGRAFÍA CONSULTADA

ANSI/NISO. 2007. The Dublin Core Metadata Element Set. Bethesda, Maryland, U.S.A : National Information Standards Organization, 2007. 15 pp. ISSN: 1041-5635.

European Software Institute. 2007. Tecnalía. Motivación para la mejora de proceso basada en CMMI. [En línea]. 2007. [Citado el: 28 de Febrero de 2012.] Disponible en: www.adictosaltrabajo.com/tutoriales/cmmlOverview/CMMI_Overview.pdf.

Faba, C y N, Nuño, M. V. 2004. La nueva gestión en las bibliotecas virtuales. [En línea] [Citado el: 25 de Noviembre de 2012.] Disponible en: www.aab.es/pdfs/baab74/74a2.pdf.

Guenther, R. 2004. New and traditional descriptive formats in the library environment. [En línea] [Citado el: 31 de Enero de 2012.] Disponible en: <http://dc2004.library.sh.cn/english/prog/ppt/ifla.ppt>.

Hodgdon, J. 2012. Drupal programming from an object-oriented perspective. [En línea]. Drupal.org. 2009. [Citado el: 1 de Mayo de 2012.] Disponible en: <http://drupal.org/node/547518>.

Jacobson, I; Booch, G y Rumbaugh, J. 2000. El Proceso Unificado de Desarrollo de Software. Madrid : Addison-Wesley, 2000. 464 pp. ISBN: 9788478290369.

jQuery. 2010. Browser Compatibility. [En línea] [Citado el: 10 de Mayo de 2012.] Disponible en: http://docs.jquery.com/Browser_Compatibility.

jQuery. 2012. License. [En línea]. jQuery Foundation, 2012. [Citado el: 10 de Mayo de 2012.] Disponible en: <http://jquery.org/license>.

Lamarca, L. M. J. 2006. Hipertexto: el nuevo concepto de documento en la cultura de la imagen. Tesis (Doctora en Fundamentos, Metodología y Aplicaciones de las Tecnologías Documentales y Procesamiento de la Información). [En línea]. Madrid, España : Universidad Complutense de Madrid. Facultad de Ciencias de la Información. Dpto. de Biblioteconomía y Documentación. 2006. 184 p. [Citado el: 9 de Diciembre de 2011] Disponible en: <http://www.hipertexto.info/documentos/indice.htm>.

Méndez, E. M. 2002. Metadatos y recuperación de información: Estándares, problemas y aplicabilidad en bibliotecas digitales. Ediciones Trea, 2002. 429 pp. Mención de serie Biblioteconomía y administración cultural no. 66. ISBN: 84-9704-055-4.

Ocampo, A. M. 2001. Manual formato MARC. Santafé de Bogotá D.C : Editorial Universitaria, 2001. 271 pp. ISBN 959-1-0083-6.

Raya, H. M. G y Zulueta, B. M. E. 2011. Textos científico-técnicos. ¿Cómo crearlos? La Habana: Científico-Técnica, 2011. 194 pp. ISBN: 978-959-05-0621-5.

VanDyk, J. K. 2008. Pro Drupal 6 Development. 667 pp. ISBN-13 (pbk): 978-1-4302-0989-8. ISBN-13 (electronic): 978-1-4302-0990-4.

GLOSARIO DE TÉRMINOS

AJAX: Java Script asíncrono y XML (AJAX acrónimo de *Asynchronous Java Script And XML*).

CSS: Hojas de Estilos en Cascada (CSS del inglés *Cascading Style Sheets*).

CMS: Sistema de Gestión de Contenidos (CMS del inglés *Content Management Systems*).

BSD: Distribución de Software Berkeley (BSD del inglés *Berkeley Software Distribution*).

DP: Proveedor de datos (DP del inglés *data provider*).

GNU/GPL: Licencia Pública General de GNU (GNU/GPL del inglés *GNU General Public License*).

HTTP: Protocolo de Transferencia de Hipertexto (HTTP del inglés *Hypertext Transfer Protocol*).

OAI-PMH: Iniciativa de Archivos Abiertos - Protocolo para la Transmisión de Metadatos (OAI-PMH del inglés *Open Archives Initiative – Protocol for Metadata Harvesting*).

PHP: Procesador de Hipertextos (PHP del inglés *Hypertext Pre-processor*).

RUP: Proceso Unificado de Rational (RUP del inglés *Rational Unified Process*).

SI: Sistemas de información.

SP: Proveedor de servicios (SP del inglés *service provider*).

TCP/IP: Protocolo de Control de Transmisión / Protocolo de Internet (TCP/IP del inglés *Transmission Control Protocol / Internet Protocol*).

W3C: Consorcio de la Web o Telaraña Mundial (W3C del inglés *World Wide Web Consortium*).

XHTML: Lenguaje de Marcado de Hipertexto Extensible (XHTML del inglés *eXtensible Hypertext Markup Language*).

XML: Lenguaje de Marcado Extensible (XML del inglés *Extensible Markup Language*).

Z39.50: Aplicación de Servicios de Definición y Especificación de Protocolo del inglés *Application Service Definition and Protocol Specification*.

ANEXOS

Anexo 1. Descripción de casos de uso del sistema

Tabla 1.1. Descripción del caso de uso Autenticar.

Objetivo	El usuario se autenticará en el sistema para que se le sean otorgados los permisos necesarios para realizar las tareas específicas de acuerdo a su rol.	
Actores	Administrador: (Inicia) Se autentica.	
Resumen	El administrador introduce su usuario y contraseña y el sistema comprueba que los datos sean correctos.	
Complejidad	Baja.	
Prioridad	Crítico.	
Precondiciones	_____	
Postcondiciones	Quedan habilitadas las funcionalidades según los privilegios que tenga.	
Flujo de eventos		
Flujo básico		
	Actor	Sistema
1.	Introduce su usuario y contraseña en el sistema de autenticación.	Comprueba que los datos estén registrados correctamente, autentica y le concede los privilegios pertinentes.
Flujos alternos		
1a. Los datos no son válidos.		
	Actor	Sistema
1.		Muestra un cartel de error y regresa al paso 1 del actor.
Requisitos no funcionales	Interfaz, seguridad.	
Asuntos pendientes	Incluir auto completamiento de usuario en el campo de usuario.	

Tabla 1.2. Descripción del caso de uso Crear documento.

Objetivo	Almacenar documentos en el sistema.	
Actores	Administrador, Reloj.	
Resumen	Después de finalizado el proceso de indexación, se crean documentos.	
Complejidad	Alta.	
Prioridad	Crítico.	
Precondiciones	El administrador se encuentra autenticado y tienen que existir documentos creados.	
Postcondiciones	Quedó creado el documento.	
Flujo de eventos		
Flujo básico		
	Actor	Sistema
1.		Culminado el proceso de indexación, el sistema almacena el o los nuevos documentos y notifica la creación.
Requisitos no funcionales	Interfaz, seguridad, estándares aplicables.	
Asuntos pendientes	_____	

Tabla 1.3. Descripción del caso de uso Listar documento.

Objetivo	Lista documentos que se encuentran en el sistema.
Actores	Administrador: (Inicia) Lista documentos.
Resumen	El administrador del sistema lista los documentos.
Complejidad	Alta.
Prioridad	Crítico.
Precondiciones	El administrador se encuentra autenticado y tienen que existir

	documentos creados.	
Postcondiciones	Quedó listado el documento.	
Flujo de eventos		
Flujo básico		
	Actor	Sistema
1.	Selecciona la opción de listar los documentos.	Lista los documentos creados.
Requisitos no funcionales	Interfaz, seguridad, estándares aplicables.	
Asuntos pendientes	_____	

Tabla 1.4. Descripción del caso de uso Modificar documento.

Objetivo	Edita documentos que se encuentran en el sistema.	
Actores	Administrador: (Inicia) Edita documentos.	
Resumen	El administrador del sistema modifica los datos de los documentos.	
Complejidad	Baja.	
Prioridad	Crítico.	
Precondiciones	El administrador se encuentra autenticado y tienen que existir documentos creados.	
Postcondiciones	Quedó modificado el documento.	
Flujo de eventos		
Flujo básico		
	Actor	Sistema
1.	Selecciona la opción de listar los documentos.	Lista los documentos creados. Ver descripción del CU-3: Listar documento.
2.	Selecciona el documento a editar.	Muestra los datos del documento.

3.	Modifica los datos y selecciona la opción de guardar.	Verifica que todos los campos requeridos estén llenos y los datos correctos.
		Almacena los cambios y notifica su modificación.
Flujos alternos		
3a. Campos vacíos o errores en los campos.		
	Actor	Sistema
1.		Muestra un mensaje de error para comunicar si hay campos requeridos vacíos y si hay errores. Volver a la acción 3 del actor.
Requisitos no funcionales		Interfaz, seguridad, estándares aplicables.
Asuntos pendientes		_____

Tabla 1.5. Descripción del caso de uso Eliminar documento.

Objetivo	Elimina documentos que se encuentran en el sistema.	
Actores	Administrador: (Inicia) Elimina documentos.	
Resumen	El administrador del sistema elimina documentos.	
Complejidad	Medio.	
Prioridad	Crítico.	
Precondiciones	El administrador se encuentra autenticado y tienen que existir documentos creados.	
Postcondiciones	Quedó eliminado el documento.	
Flujo de eventos		
Flujo básico		
	Actor	Sistema

1.	Selecciona la opción de listar los documentos.	Lista los documentos creados. Ver descripción del CU-3: Listar documento.
2.	Selecciona el documento a eliminar.	Elimina el documento y notifica su eliminación. Volver a la acción 1 del sistema.
Requisitos no funcionales		Interfaz, seguridad, estándares aplicables.
Asuntos pendientes		_____

Tabla 1.6. Descripción del caso de uso Gestionar fuente.

Objetivo	Operar con las fuentes ya sea para crear, listar, modificar o eliminarlas.	
Actores	Administrador: (Inicia) Adiciona, ve, modifica y elimina las fuentes.	
Resumen	El administrador gestiona las fuentes para garantizar el proceso de guardar las colecciones para futura indexación.	
Complejidad	Alta.	
Prioridad	Crítico.	
Precondiciones	El administrador se encuentra autenticado.	
Postcondiciones	Quedó creada, mostrada, modificada o eliminada la o las fuentes.	
Flujo de eventos		
Flujo básico		
	Actor	Sistema
1.	Se dirige al menú de las fuentes.	Muestra las opciones que se pueden hacer con las fuentes: Crear fuente, Listar fuente, Modificar fuente y Eliminar fuente.
2.	Selecciona una de las opciones.	Permite según la opción: - Adicionar una nueva fuente.

		<p>Ver Sección 1: “Crear fuente”.</p> <ul style="list-style-type: none"> - Listar las fuentes. Ver Sección 2: “Listar fuente”. - Modificar una fuente. Ver Sección 3: “Modificar fuente”. - Eliminar una fuente. Ver Sección 4: “Eliminar fuente”.
Sección 1: “Crear fuente”		
Flujo básico		
	Actor	Sistema
1.	Introduce los datos de la fuente y presiona la opción de guardar.	Verifica que todos los campos requeridos estén llenos, los datos correctos, que sea una fuente válida y que contenga el estándar de metadatos “oai_dc”.
2.		Almacena la nueva fuente y notifica su creación.
Flujos alternos		
1a. Campos vacíos o errores en los campos.		
	Actor	Sistema
1.		<p>Muestra un mensaje de error emitiendo si hay campos requeridos vacíos, si hay errores, si no es una fuente válida o si no contiene el estándar de metadatos “oai_dc”.</p> <p>Volver a la acción 1 del actor.</p>
Sección 2: “Listar fuente”		
Flujo básico		

	Actor	Sistema
1.		Muestra las fuentes creadas.
Sección 3: “Modificar fuente”		
Flujo básico		
	Actor	Sistema
1.		Muestra las fuentes. Ver sección 2.
2.	Selecciona la fuente a editar.	Muestra los datos de la fuente.
3.	Modifica los datos y selecciona la opción de guardar.	Verifica que todos los campos requeridos estén llenos y que no tengan errores.
		Almacena los cambios y notifica su modificación.
Flujos alternos		
3a. Campos vacíos o errores en los campos.		
	Actor	Sistema
1.		Muestra un mensaje de error emitiendo si hay campos requeridos vacíos y si hay errores. Volver a la acción 3 del actor.
Sección 4: “Eliminar fuente”		
Flujo básico		
	Actor	Sistema
1.		Muestra las fuentes. Ver sección 2.
2.	Selecciona la fuente a eliminar.	Elimina la fuente y notifica su eliminación. Volver a la acción 1 del sistema.
Requisitos no funcionales		Interfaz, seguridad, estándares aplicables.

Asuntos pendientes	_____
---------------------------	-------

Tabla 1.7. Descripción del caso de uso Gestionar colección.

Objetivo	Operar con las colecciones ya sea para crear, listar, modificar o eliminarlas.	
Actores	Administrador: (Inicia) Adiciona, ve, modifica y elimina las colecciones.	
Resumen	El administrador gestiona las colecciones para garantizar el proceso de indexado.	
Complejidad	Alta.	
Prioridad	Crítico.	
Precondiciones	El administrador se encuentra autenticado.	
Postcondiciones	Quedó creada, mostrada, modificada o eliminada la o las colecciones.	
Flujo de eventos		
Flujo básico		
	Actor	Sistema
1.	Se dirige al menú de colecciones.	Muestra las opciones que se pueden hacer: Crear colección, Listar colección, Modificar colección y Eliminar colección.
2.	Selecciona una de las opciones.	Permite según la opción: <ul style="list-style-type: none"> - Adicionar una nueva colección. Ver Sección 1: "Crear colección". - Listar las colecciones. Ver Sección 2: "Listar colección". - Modificar una colección. Ver

		<p>Sección 3: “Modificar colección”.</p> <ul style="list-style-type: none"> - Eliminar una colección. Ver Sección 4: “Eliminar colección”.
Sección 1: “Crear colección”		
Flujo básico		
	Actor	Sistema
1.	Selecciona una fuente y presiona el botón de aceptar.	Verifica que se haya seleccionado una fuente y consulta las colecciones que contiene dicha fuente.
2.		Muestra las colecciones que pertenecen a la fuente consultada.
3.	Marca las colecciones que desee adicionar e introduce el tiempo de actualización de las indexaciones de cada uno de ellos y presiona el botón adicionar.	Verifica que exista al menos una colección seleccionada.
4.		Almacena el o las colecciones y notifica su almacenamiento.
Flujos alternos		
1a. Campo no seleccionado.		
	Actor	Sistema
1.		<p>Muestra un mensaje de error emitiendo que no se ha seleccionado ninguna fuente.</p> <p>Volver a la acción 1 del actor.</p>
1b. Colecciones no seleccionadas.		
	Actor	Sistema
1.		Muestra un mensaje de error emitiendo que no se ha seleccionado

		ninguna colección. Volver a la acción 3 del actor.
Sección 2: “Listar colección”		
Flujo básico		
	Actor	Sistema
1.		Muestra las colecciones almacenadas en el sistema.
Sección 3: “Modificar colección”		
Flujo básico		
	Actor	Sistema
1.		Muestra las colecciones. Ver sección 2.
2.	Actualiza el tiempo de actualización de la colección que desee y presiona el botón de actualizar.	Almacena los cambios y notifica su modificación.
Sección 4: “Eliminar colección”		
Flujo básico		
	Actor	Sistema
1.		Muestra las colecciones. Ver sección 2.
2.	Selecciona la o las colecciones a eliminar y presiona la opción de eliminar.	Elimina la o las colecciones y notifica la eliminación. Volver a la acción 1 del sistema.
Requisitos no funcionales		Interfaz, seguridad, estándares aplicables.
Asuntos pendientes		_____

Tabla 1.8. Descripción del caso de uso Actualizar token.

Objetivo	Renovar la fecha de actualización.	
Actores	Administrador, Reloj.	
Resumen	Terminado el flujo de indexación de documentos, se actualiza el campo de última fecha de indexación de una o varias colecciones y/o fuentes.	
Complejidad	Alta.	
Prioridad	Crítico.	
Precondiciones	_____	
Postcondiciones	Realizar el proceso de indexado de documentos.	
Flujo de eventos		
Flujo básico		
	Actor	Sistema
1.		Actualiza el campo de última fecha de actualización de las colecciones y/o fuentes de las que se hayan indexado documentos.
Requisitos no funcionales	Seguridad, estándares aplicables.	
Asuntos pendientes	_____	

Tabla 1.9. Descripción del caso de uso Indexar documento desde archivo.

Objetivo	Crear nodos de Drupal representando a los documentos provenientes de un archivo con extensión XML.
Actores	Administrador (Inicia): Indexa documentos a partir del archivo en formato XML.
Resumen	Se interpreta el fichero XML y se indexa uno o varios documentos.
Complejidad	Alta.

Prioridad	Crítico.	
Precondiciones	_____	
Postcondiciones	Quedó creado un documento.	
Flujo de eventos		
Flujo básico		
	Actor	Sistema
1.	Selecciona la opción de indexar desde archivo local.	Muestra los datos a llenar.
2.	Completa los datos.	Verifica que todos los campos requeridos estén llenos y no contengan errores.
3.		Indexa los contenidos.
Flujos alternos		
1a. Campos vacíos o errores en los campos.		
	Actor	Sistema
1.		Muestra un mensaje de error emitiendo que faltan datos o que existe algún error. Volver a la acción 1 del actor.
Requisitos no funcionales	Interfaz, seguridad, estándares aplicables.	
Asuntos pendientes	_____	

Anexo 2. Diagrama de clases del análisis

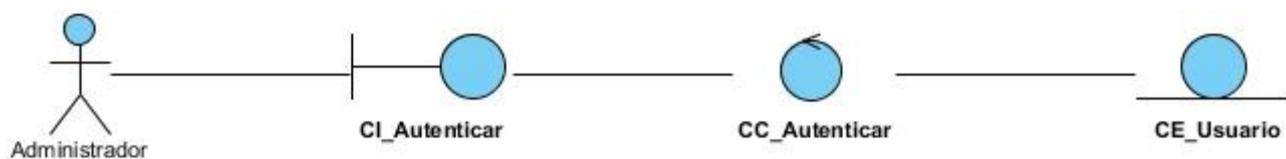


Fig. 2.1: Diagrama de clases del análisis: CU-1 Autenticar.



Fig. 2.2: Diagrama de clases del análisis: CU-3 Listar documento.



Fig. 2.3: Diagrama de clases del análisis: CU-4 Modificar documento.



Fig. 2.4: Diagrama de clases del análisis: CU-5 Eliminar documento.

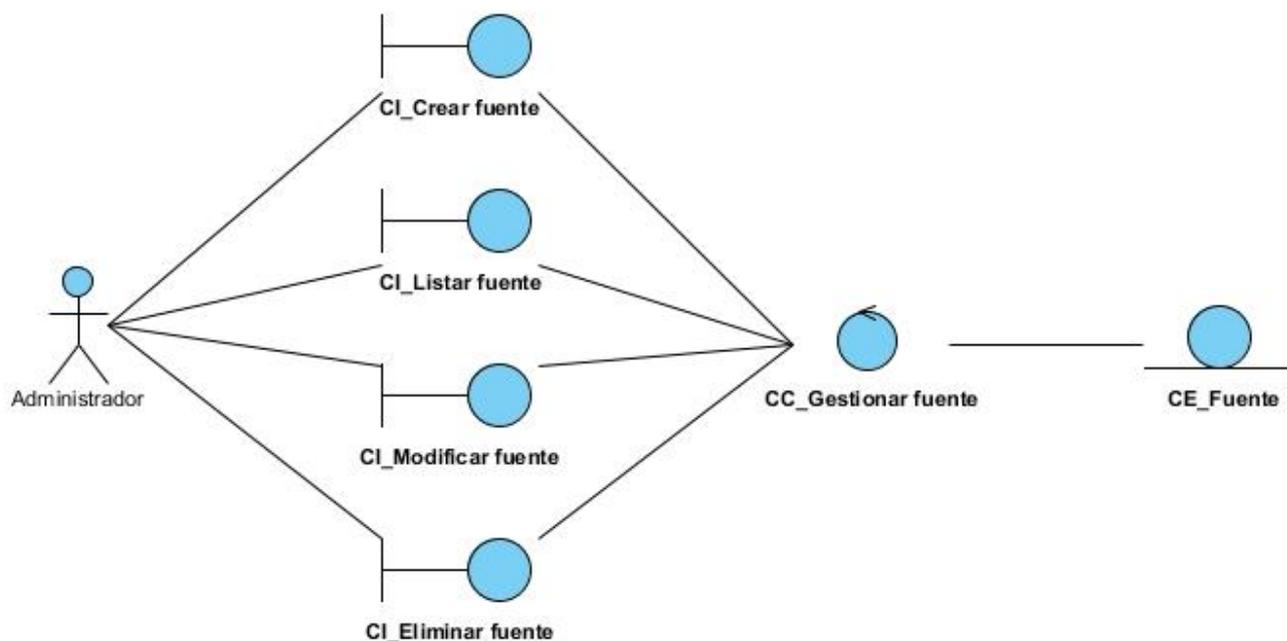


Fig. 2.5: Diagrama de clases del análisis: CU-6 Gestionar fuente.

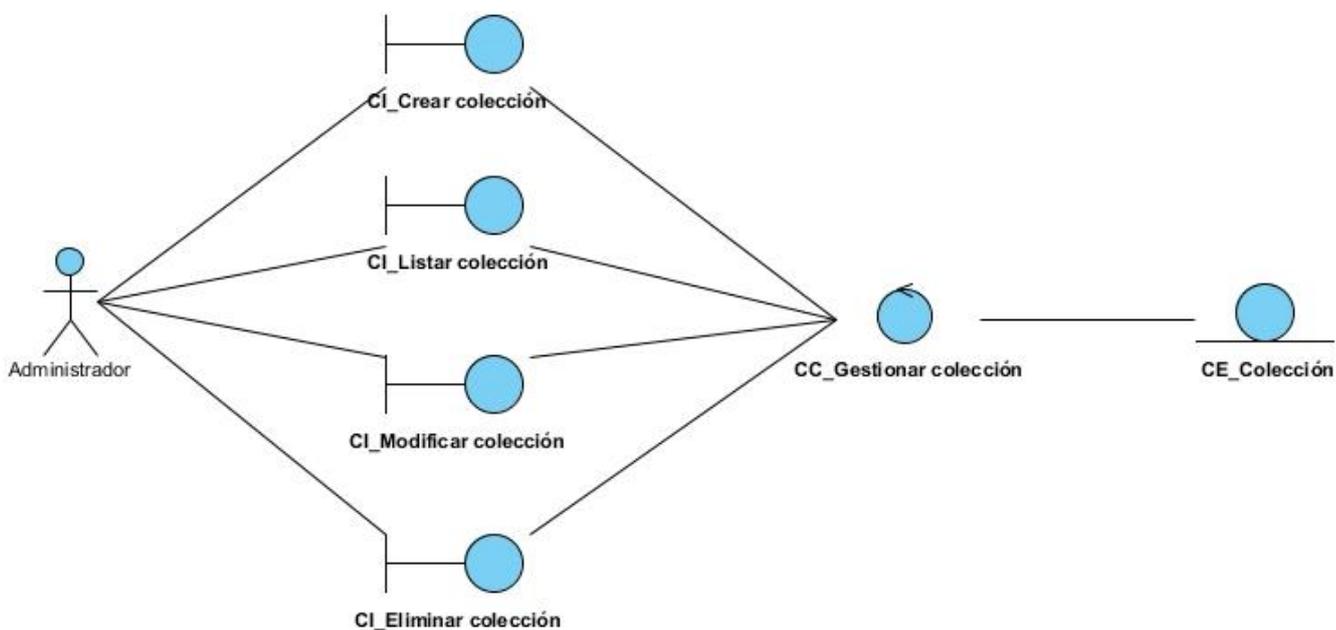


Fig. 2.6: Diagrama de clases del análisis: CU-7 Gestionar colección.

Anexo 3. Descripción de las tablas

Tabla 3.1. Descripción de la tabla *content_type_coleccion*.

Nombre: content_type_coleccion		
Descripción: Almacena los datos de las colecciones.		
Atributo	Tipo	Descripción
vid	integer	Identificador de la versión
nid	integer	Identificador del nodo.
field_coleccion_actualizacion_value	timestamp	Fecha y hora de última indexación.
field_coleccion_fuente_nid	integer	Identificador de la fuente.
field_coleccion_periodicidad_value	text	Período de actualización.
field_coleccion_setspec_value	text	Identificador de la colección.

Tabla 3.2. Descripción de la tabla *content_type_fuente*.

Nombre: content_type_fuente		
Descripción: Almacena los datos de las fuentes.		
Atributo	Tipo	Descripción
vid	integer	Identificador de la versión
nid	integer	Identificador del nodo.
field_fuente_urlbase_value	text	Url Base.
field_fuente_periodicidad_value	text	Período de actualización.
field_fuente_firstdate_value	text	Primera fecha de aparición.
field_fuente_dateformat_value	text	Formato de fecha.
field_fuente_granulate_value	text	Granularidad de la fecha.
field_fuente_actualizacion_value	timestamp	Fecha y hora de última indexación.
field_fuente_format_value	text	Formato de metadato.

Tabla 3.3. Descripción de la tabla *content_type_documento*.

Nombre: content_type_documento		
Descripción: Almacena los datos de los documentos.		
Atributo	Tipo	Descripción
vid	integer	Identificador de la versión
nid	integer	Identificador del nodo.
field_doc_description_value	text	Descripción del documento.
field_doc_language_value	text	Lenguaje del documento.
field_doc_relation_nid	integer	Relación con otros documentos.
field_doc_coverage_value	text	Colaboradores del documento.
field_doc_creator_value	text	Creador del documento.
field_doc_publisher_value	text	Editor.
field_doc_rights_value	text	Derechos del documento.
field_doc_date_value	integer	Fecha de indexación.
field_doc_popup_date_value	charactervarying	Lenguaje del documento.
field_doc_type_value	text	Tipo de documento.
field_doc_format_value	text	Formato del documento.
field_doc_own_value	text	Propio.
field_doc_visits_value	integer	Contador de visitas.
field_doc_source_nid	integer	Identificador de la fuente.

Tabla 3.4. Descripción de la tabla *content_field_doc_contributer*.

Nombre: content_field_doc_contributer		
Descripción: Almacena los contribuidores al desarrollo de un documento.		
Atributo	Tipo	Descripción
vid	integer	Identificador de la versión
nid	integer	Identificador del nodo.
delta	integer	Número de contribuidor almacenado.
field_doc_contributer_value	text	Nombre del contribuidor.

Tabla 3.5. Descripción de la tabla *content_field_doc_identifier*.

Nombre: content_field_doc_identifier		
Descripción: Almacena los identificadores de un documento.		
Atributo	Tipo	Descripción
vid	integer	Identificador de la versión
nid	integer	Identificador del nodo.
delta	integer	Número del identificador almacenado.
field_doc_identifier_value	text	Identificador del documento.

Tabla 3.6. Descripción de la tabla *content_field_doc_subject*.

Nombre: content_field_doc_subject		
Descripción: Almacena los temas de un documento.		
Atributo	Tipo	Descripción
vid	integer	Identificador de la versión
nid	integer	Identificador del nodo.
delta	integer	Número del tema almacenado.
field_doc_subject_value	text	Tema del documento.

Tabla 3.7. Descripción de la tabla *content_field_doc_file*.

Nombre: content_field_doc_file		
Descripción: Almacena los ficheros adjuntos de un documento.		
Atributo	Tipo	Descripción
vid	integer	Identificador de la versión
nid	integer	Identificador del nodo.
delta	integer	Número del tema almacenado.

Anexo 4. Diseño de casos de prueba basadas en casos de uso

Tabla 4.1. Caso de prueba del escenario Autenticar del CU-1.

Escenario	Descripción	Usuario	Contraseña	Respuesta del sistema	Flujo central
EC 1.1 [Usuario correcto]	Datos válidos para un usuario que se autentica en el sistema.	V	V	El usuario queda autenticado en el sistema.	El usuario ingresa su nombre de usuario y contraseña en el bloque de autenticación que se encuentra en la zona izquierda del sitio.
		admin	Admin OAIPMH		
EC 1.2 [Usuario incorrecto]	Datos no válidos para un usuario que se autentica en el sistema.	I	V	La aplicación muestra un mensaje de error y por lo tanto el usuario no queda autenticado.	
			dss \$% ,F		
		V	I		
		harvester			
		I	V		
	proveedor	HarvesteR #			

Tabla 4.2. Caso de prueba del escenario Crear fuente del CU-7.

Escenario	Descripción	Título	Url Base	Respuesta del sistema	Flujo central
EC 1.1 [Fuente correcta]	Datos válidos para una fuente que es creada.	V	V	La fuente queda creada.	El usuario con permisos para administrar el proveedor de servicios presiona la opción de "Fuentes OAI" en el bloque "Proveedor de Servicios" y luego y en la pestaña "Agregar fuente" e introduce los datos.
		Repositorio UCI	http://repositorio_institucional.uci.cu/oai/request		
EC 1.2 [Fuente incorrecta]	Datos no válidos para una fuente que intenta ser creada.	I	I	La aplicación muestra un mensaje de error. "La fuente no es válida porque no implementa el protocolo OAI-PMH."	
		V	I		
		Repositorio UCI			
		V	I		
	Repositorio UCI	http://intranet2.uci.cu			

Tabla 4.3. Caso de prueba del escenario Crear colección del CU-8.

Escenario	Descripción	Fuente	Colección	Respuesta del sistema	Flujo central
EC 1.1 [Colección correcta]	Datos válidos para una colección que es creada.	V	V	La colección queda creada.	El usuario con permisos para administrar el proveedor de servicios presiona la opción de "Colecciones" en el bloque "Proveedor de Servicios" y luego y en la pestaña "Agregar colecciones".
		Repositorio UCI	Tesis, RCCI, Publicaciones		
EC 1.2 [Colección incorrecta]	Datos no válidos para una colección que intenta ser creada.	I	I	La aplicación muestra un mensaje de error, mostrando la razón por la que no se pudo crear la colección.	
		V	I		
		Repositorio UCI			
		V	V		
		Repositorio UCI	Maestría		

Tabla 4.4. Caso de prueba del escenario Modificar colección del CU-8.

Escenario	Descripción	Fuente	Colección	Identidad	Reanudación	Respuesta del sistema	Flujo central
EC 1.1 [Colección correcta]	Datos válidos para una colección que es modificada.	V	V	V	V	La colección queda modificada.	El usuario con permisos para administrar el proveedor de servicios presiona la opción de "Colecciones" en el bloque "Proveedor de Servicios" y luego y en la pestaña "Colecciones" y edita una colección.
		Repositorio UCI	Tesis	hdl_ident_3748			
EC 1.2 [Colección incorrecta]	Datos no válidos para una colección que intenta ser modificada.	I	I	I	V	La aplicación muestra un mensaje de error, mostrando la razón por la que no se pudo modificar la colección.	
		V	I	V	V		
		Repositorio UCI		hdl_ident_3748	2001-01-01T00:00:00Z		
		V	V	I	V		
		Repositorio UCI	Maestría		2001-01-01T00:00:00Z		

Tabla 4.5. Caso de prueba del escenario Indexar documento desde archivo del CU-12.

Escenario	Descripción	Fuente	Tipo de recurso	XML	Respuesta del sistema	Flujo central
EC 1.1 [Datos correctos]	Datos válidos para indexar documento desde un archivo local.	V	V	V	Se indexó un número determinado de documentos.	El usuario con permisos para administrar el proveedor de servicios presiona la opción de "Documentos" en el bloque "Proveedor de Servicios" y luego y en la pestaña "Indexar desde archivo local" e introduce los datos.
		Repositorio UCI	Revistas	citeceer.xml		
EC 1.2 [Datos incorrectos]	Datos no válidos para indexar documento desde un archivo local.	I	I	I	La aplicación muestra un mensaje de error, mostrando que hay datos incorrectos.	
		Seleccione	Seleccione			
		V	V	I		
		Repositorio UCI	Informes	source.html		
		V	V	V		
Repositorio UCI	Ponencia	citeceer.xml				

Anexo 5. No conformidades

Tabla 5.1. No conformidades.

Iteración	Elemento	#	Descripción	Ubicación	Estado
1	Aplicación	1	El mensaje que se muestra cuando no se introdujo el usuario: “El campo Usuario es obligatorio.” en el bloque de autenticación tiene que ser de color rojo.	Bloque de autenticación.	Resuelta
		2	Cuando se va a crear una fuente, el campo para introducir el nombre de la fuente dice “Fuentes” cuando debería ser “Fuente”.	Sección de Fuentes.	
		3	Cuando se crea una fuente con una url que no cumple con las características del protocolo OAI-PMH, se muestra un mensaje: “Fuente no válida.”, cuando debería dar más detalles: “La fuente no es válida porque no implementa el protocolo OAI-PMH.”.	Pestaña de Agregar fuente dentro de la sección Fuentes.	
		4	Le falta la tilde a la palabra “Colección”.	Listado de colecciones de una fuente en la pestaña de Agregar colección.	
		5	No se crea el rol de administrador del proveedor cuando se instala el módulo.	Roles.	
		6	El bloque de gestión del proveedor de servicios sólo se le tiene que mostrar a los usuarios con rol de administrador que provee el módulo.	Bloque Proveedor de Servicios.	
		7	La recolección de documentos hay que dividirla en dos secciones, “Por fuente” y “Por colección”.	Sección de Documentos.	
		8	Cuando se agrega una nueva fuente, se muestra un mensaje confirmándolo, al que le falta el punto final.	Pestaña de Agregar fuente dentro de la sección Fuentes.	

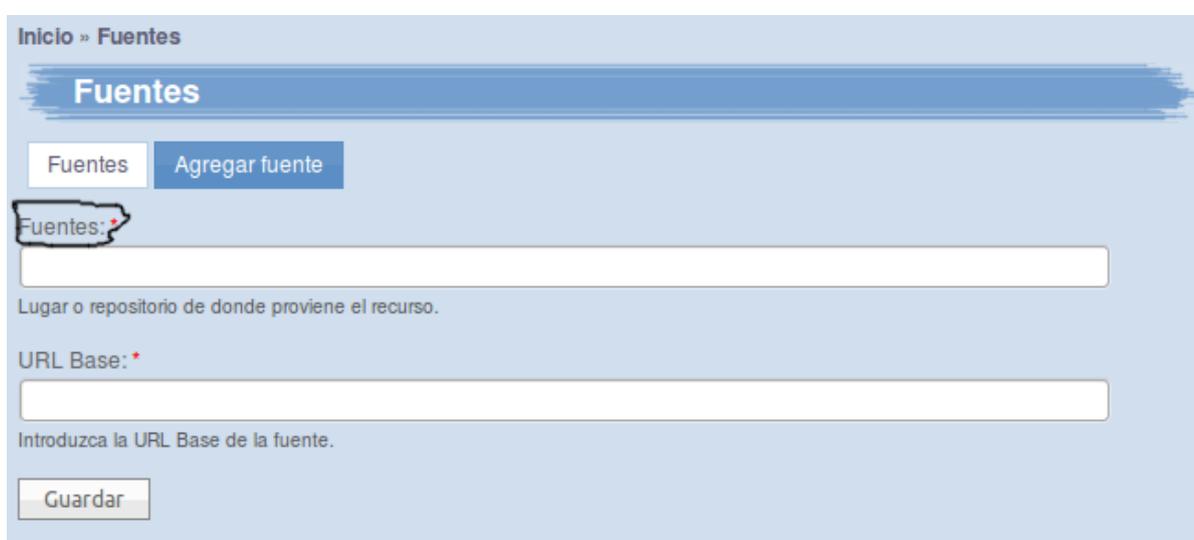
2	1	Quando se procede a crear una colección hay que seleccionar una fuente y luego se presiona el botón de Mostrar. Luego de esto se muestra un listado de colecciones de la fuente escogida. El nombre del listado debe indicar la fuente a la que pertenecen las colecciones por ejemplo: “Listado de colecciones de la fuente Repositorio UCI”, cuando en realidad dice: “Listado de colecciones de la fuente”.	Agregar colección dentro de la sección de Colecciones.
	2	Quando el administrador se dirige a crear una colección, el flujo a seguir descrito en el caso de prueba correspondiente, no se cumple en la aplicación.	Agregar colección dentro de la sección de Colecciones.
	3	Se muestran campos innecesarios.	Editar una fuente.
	4	Se muestran campos innecesarios.	Editar documento.
	5	Quando se indexa automáticamente, falta mostrar la información de los documentos almacenados ya sea de una colección como de una fuente.	Ejecución de cron.
3	1	Al indexar desde un archivo local hay que llenar todos los campos. En caso de faltar alguno debe mostrarse un mensaje indicándolo. Cuando el XML se queda vacío no se muestra ningún mensaje cuando debería mostrar: “El campo XML es obligatorio.”.	Pestaña Indexar desde archivo local en la sección de Documentos.
	2	Al indexar desde un archivo local si se sube un fichero que no sea XML, se muestra un mensaje con error ortográfico en la palabra “extensión”.	Pestaña Indexar desde archivo local en la sección de Documentos.
	3	El listado de documentos tiene que mostrar el estado de publicación.	Pestaña Documentos en la sección de Documentos.

Ejemplo de no conformidades detectadas:



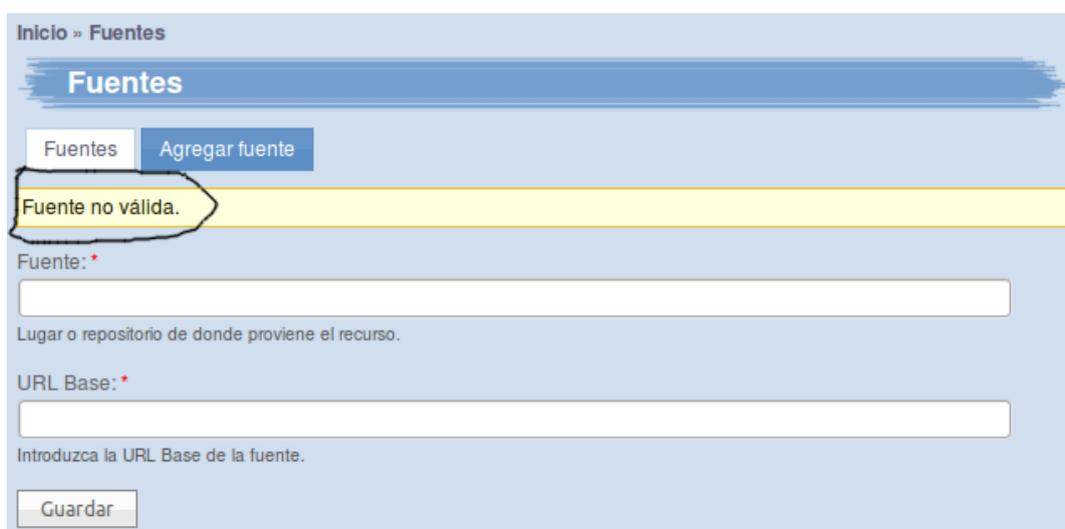
The screenshot shows a login form titled "Inicio de sesión". It has two input fields: "Usuario:" and "Contraseña:". The "Usuario:" field is empty and has a red border, with a red error message box above it that says "El campo Usuario es obligatorio." To the right of the form, there is a blue header area with the text "¡Bienvenido a tu nuevo sitio Drupal!" and "Por favor sigue estos pasos para configurar y comenzar a usar tu sitio:".

Fig. 5.1. No conformidad 1. Iteración 1.



The screenshot shows the "Inicio » Fuentes" page. The main heading is "Fuentes". There are two buttons: "Fuentes" and "Agregar fuente". Below the buttons is a form with two input fields: "Fuentes:" and "URL Base:". The "Fuentes:" field is empty and has a red error message box above it that says "Fuentes:". Below the "URL Base:" field is a "Guardar" button.

Fig. 5.2. No conformidad 2. Iteración 1.



The screenshot shows the "Inicio » Fuentes" page. The main heading is "Fuentes". There are two buttons: "Fuentes" and "Agregar fuente". Below the buttons is a form with two input fields: "Fuente:" and "URL Base:". The "Fuente:" field is empty and has a yellow error message box above it that says "Fuente no válida." Below the "URL Base:" field is a "Guardar" button.

Fig. 5.3. No conformidad 3. Iteración 1.



Fig. 5.4. No conformidad 1. Iteración 2.

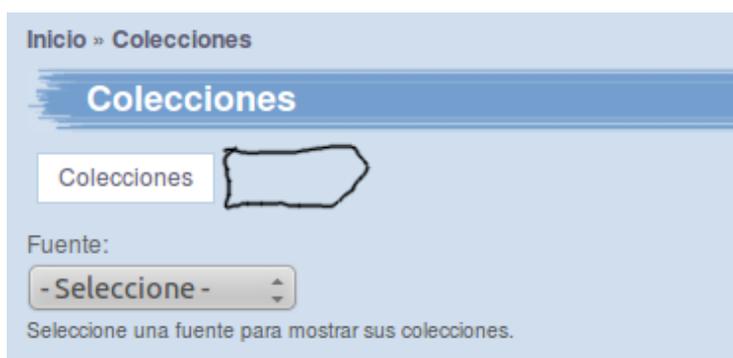


Fig. 5.5. No conformidad 2. Iteración 2.

Documentos Indexar desde archivo local Indexar documentos

- Debe seleccionar una *Fuente*.
- Debe seleccionar un *Tipo de recurso* predominante en la fuente a indexar.

Fuente: *
- Selecciona -
Lugar o repositorio de donde proviene el recurso.

Tipo de recurso: *
- Selecciona -
Tipo de recurso predominante en la fuente a indexar.

XML: *

Archivo XML con datos en formato OAI. Examinar...

Indexar

Fig. 5.6. No conformidad 1. Iteración 3.

Documentos

Documentos Indexar desde archivo local Indexar documentos

El archivo tiene que presentar extensión `xml`.

Fuente: *
Repositorio UCI
Lugar o repositorio de donde proviene el recurso.

Tipo de recurso: *
Diaporama
Tipo de recurso predominante en la fuente a indexar.

XML: *

Archivo XML con datos en formato OAI. Examinar...

Indexar

Fig. 5.7. No conformidad 2. Iteración 3.

Anexo 6. Vistas del sistema



Fig. 6.1. Listado de fuentes.



Fig. 6.2. Adicionar una fuente.

Colecciones

Colecciones

Fuente:

Operaciones:

<input type="checkbox"/>	Colección ▲	Fecha	Publicado		
0 elementos.		<input type="button" value="Cancelar la selección"/>			
<input type="checkbox"/>	Evento Juvenil Martiano	29/03/2012	Sí	editar	eliminar
<input type="checkbox"/>	Eventos	29/03/2012	Sí	editar	eliminar
<input type="checkbox"/>	Fórum de Ciencia y Técnica	10/05/2012	Sí	editar	eliminar
<input type="checkbox"/>	Jornada Científica Estudiantil	29/03/2012	Sí	editar	eliminar
<input type="checkbox"/>	Mi Web por Cuba	24/04/2012	Sí	editar	eliminar
<input type="checkbox"/>	Otras Publicaciones	03/04/2012	Sí	editar	eliminar
<input type="checkbox"/>	Peña Tecnológica	04/04/2012	Sí	editar	eliminar
<input type="checkbox"/>	Publicaciones	29/03/2012	Sí	editar	eliminar
<input type="checkbox"/>	RCCI	29/03/2012	Sí	editar	eliminar

Fig. 6.3. Listado de colecciones.

Colecciones

Colecciones Agregar colecciones

Fuente:
- Seleccione -

Seleccione una fuente para mostrar sus colecciones.

Mostrar

Listado de colecciones de la fuente *Repositorio UCI*:

- Evento Juvenil Martiano
- Fórum de Ciencia y Técnica
- Fórum de Historia
- Jornada Científica Estudiantil
- Mi Web por Cuba
- Peña Tecnológica
- Publicaciones
- RCCI
- Serie Científica
- Tesis de Doctorado
- Tesis de Maestría
- Trabajos de Diploma
- UCIENCIA 2012
- UXI
- Eventos
- Otras Publicaciones
- Revistas
- Tesis
- UCIENCIA

Almacenar

Fig. 6.4. Agregar colecciones.

Documentos

Documentos | Indexar desde archivo local | Indexar documentos

Fuente: Documento:

Operaciones:

<input type="checkbox"/> Documento ▲	Fecha	Publicado	
0 elementos. <input type="button" value="Cancelar la selección"/>			
<input type="checkbox"/> ACCESIBILIDAD EN SITIOS WEB.	06/04/2012	Sí	editar eliminar
<input type="checkbox"/> ACCIONES PRÁCTICAS PARA GESTIONAR EL CONOCIMIENTO EN LA PRODUCCIÓN DE SOFTWARE	06/04/2012	Sí	editar eliminar
<input type="checkbox"/> ACCIONES QUE CONTRIBUYEN AL DESARROLLO COMUNITARIO EN ZONAS CARACTERIZADAS A TRAVÉS DE ACTIVIDADES EXTENSIONISTAS.	27/04/2012	Sí	editar eliminar
<input type="checkbox"/> ACERCAMIENTO A LA ECONOMÍA DEL CONOCIMIENTO PARA LA CREACIÓN DE ORGANIZACIONES DE ALTO DESEMPEÑO (OAD).	06/04/2012	Sí	editar eliminar
<input type="checkbox"/> ActionScript. Curso optativo.	27/04/2012	Sí	editar eliminar
<input type="checkbox"/> ACTIVIDADES BASADAS EN ESTÍMULOS MUSICALES PARA DESARROLLAR LA COMPRENSIÓN AUDITIVA EN IDIOMA INGLÉS	06/04/2012	Sí	editar eliminar
<input type="checkbox"/> ACTUALIDAD DE LOS MÉTODOS PARA LA EVALUACIÓN DE PROCESOS DE DESARROLLO DE SOFTWARE.	06/04/2012	Sí	editar eliminar
<input type="checkbox"/> ACTUALIZACIÓN DEL CURSO SOBRE RECURSOS MULTIMEDIA EN FUNCIÓN DE LA DOCENCIA UNIVERSITARIA EN LA MODALIDAD SEMIPRESENCIAL	06/04/2012	Sí	editar eliminar
<input type="checkbox"/> Actualizacion y ajuste del modificador de animaciones de huesos para juegos CNEURO.	27/04/2012	Sí	editar eliminar
<input type="checkbox"/> ACTUALÍZATE, SISTEMA DE SINDICACIÓN RSS, DE CANALES DE LA UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS, NACIONALES E INTERNACIONALES	06/04/2012	Sí	editar eliminar
<input type="checkbox"/> Adaptacion de la Metodología Diseño y Desarrollo de Interfaz de Usuario Web para aplicarla al proyecto Contenidos Educativos Digitales	06/04/2012	Sí	editar eliminar
<input type="checkbox"/> Adaptacion del Modulo de Circulacion del Sistema Integrado de Gestion Bibliotecaria Koha a la Biblioteca Nacional de Cuba "Jose Martí"	06/04/2012	Sí	editar eliminar

1 2 3 4 5 6 7 8 9 ... siguiente > última »

Fig. 6.5. Listado de documentos.

Documentos

Documentos | **Indexar desde archivo local** | Indexar documentos

Fuente: *
- Seleccione -
Lugar o repositorio de donde proviene el recurso.

Tipo de recurso: *
- Seleccione -
Tipo de recurso predominante en la fuente a indexar.

XML: *
 Examinar...
Archivo XML con datos en formato OAI.

Indexar

Fig. 6.6. Indexar documentos desde archivo local.

Documentos

Documentos | Indexar desde archivo local | **Indexar documentos**

Por fuente | Por colección

Fuente:
 Filtrar

Fuente ▲	Fecha	Publicado	
Repositorio UCI	23/03/2012	Sí	indexar documentos

Fig. 6.7. Indexar documentos por fuente.

The screenshot shows a web interface titled "Documentos". At the top, there are three buttons: "Documentos", "Indexar desde archivo local", and "Indexar documentos". Below these are two tabs: "Por fuente" and "Por colección". The "Por colección" tab is selected. Underneath, there is a "Fuente:" label and a dropdown menu currently showing "- Seleccione -". Below the dropdown is the text "Seleccione una fuente para mostrar sus colecciones." and a "Mostrar" button. The main content area is titled "Listado de colecciones de la fuente *Repositorio UCI*:" and contains a list of 15 items, each with an unchecked checkbox: "Publicaciones", "RCCI", "UCIENCIA 2012", "Eventos", "Evento Juvenil Martiano", "Tesis", "Jornada Científica Estudiantil", "Tesis de Maestría", "Otras Publicaciones", "Peña Tecnológica", "Revistas", "Mi Web por Cuba", "UCIENCIA", and "Fórum de Ciencia y Técnica". At the bottom of this list is an "Indexar" button.

Fig. 6.8. Indexar documentos por colección.