BIBLIOTECA DIGITAL NUEVA GENERACIÓN LUCENE

Implementación de indexación, búsquedas y recuperación de documentos en texto completo para Bibliotecas Digitales

Cristian Ríos Villada

John Jairo Silva Zuluaga

Proyecto final para optar por el grado de Ingeniero de Sistemas

Universidad EAFIT

24 de noviembre de 2008

BIBLIOTECA DIGITAL NUEVA GENERACION LUCENE

Implementación de indexación, búsquedas y recuperación de documentos en texto completo para Bibliotecas Digitales

CRISTIAN RÍOS VILLADA

JOHN JAIRO SILVA ZULUAGA

Asesor:

Profesor EDWIN MONTOYA

Proyecto final para optar por el grado de Ingeniero de Sistemas

Medellín

Universidad Eafit

Facultad De Ingeniería de Sistemas

24 de Noviembre de 2008

A nuestras familias y amistades las cuales nos brindaron su apoyo incondicional para estar más cerca de nuestras metas profesionales.

AGRADECIMIENTOS

Los autores expresan su agradecimiento a:

Edwin Montoya, asesor y guía del presente proyecto.

Cástulo Ramírez, por su predisposición permanente e incondicional en aclarar dudas y por sus substanciales sugerencias durante el desarrollo del proyecto.

Los profesores de la Universidad, formadores técnicos y humanos, por dedicarse a incrementar nuestros conocimientos y capacidad intelectual.

Todos aquellos quienes de una u otra forma nos brindaron su apoyo durante nuestra formación profesional: A Dios, familia, amigos y compañeros de Universidad.

Muchísimas gracias!

TABLA DE CONTENIDO

LISTA DE FIGURAS

LISTA DE TABLAS

LISTA DE ANEXOS

1. Manual Técnico del sistema BDNG Lucene

2. Manual de Usuario del sistema BDNG Lucene

1. INTRODUCCIÓN

Las Bibliotecas Digitales (BD) se están convirtiendo en el principal medio para acceso a información digital de una manera organizada y estructurada. Una BD contiene datos, representados en el contenido en sí mismo, y metadatos, la forma de organizar la información.

De acuerdo a la globalización y a los cambios en las Tecnologías de Información y Comunicaciones (TIC), en el mundo se ha estado realizando proyectos que permitan generar conocimiento más conocimiento. Mucho de este conocimiento queda en el anonimato debido a que no se tiene un mecanismo que pueda indexar esta información para que posteriormente cualquier usuario en el mundo pueda consultar esta producción intelectual.

La Biblioteca de Nueva Generación (BDNG), implementada por el área de Telemática, actualmente cuenta con un mecanismo de búsqueda, creación, obtención y catalogación de información en los metadatos descriptivos de los documentos que este contiene, pero BDNG por ahora no tiene implementado los mecanismos para la búsqueda, recuperación y creación de información en los documentos mismo siendo así, difícil e ineficiente el encontrar la información deseada dentro de los documentos mismos.

Es por eso que con este proyecto se crea un módulo con el fin de hacer más fácil la recopilación y la obtención de la información contenida en los documentos guardados en BDNG, este módulo está basado en el motor de indexación LUCENE, que es un proyecto OpenSource para la indexación y recuperación de documentos.

Esta herramienta serviría a la comunidad estudiantil de la Universidad EAFIT ya que no cuenta con un sitio Web donde los estudiantes puedan guardar sus documentos y compartirlos con los demás estudiantes y a quienes puedan interesar. Por ejemplo, serviría para que las Tesis o Proyectos de Grado sean guardadas digitalmente para que así los grupos de investigación y los mismos estudiantes puedan buscar y leer la documentación digitalmente.

Con este proyecto se pretende dar respuesta a la necesidad que se tiene por parte de la comunidad académica de EAFIT de tener una biblioteca digital que en este caso en particular se encargue de gestionar los documentos digitales y su contenido.

2. OBJETIVOS

GENERAL:

Desarrollar el módulo de indexación, búsqueda y recuperación de documentos en texto completo para la Biblioteca Digital de Nueva Generación (BDNG).

ESPECÍFICOS:

- Implementar el módulo de indexación de documentos en texto completo a partir del contenido digital en BDNG, dicha indexación se realizará en línea (cada vez que se carga un documento al repositorio) o en lote.

- Implementar el módulo de búsqueda en texto completo para BDNG, el cual se debe integrar con el módulo actual de búsquedas de metadatos en BDNG.

3. MARCO TEORICO

El objetivo de este proyecto es crear una biblioteca digital de nueva generación. Es decir, donde se puedan almacenar, indexar y buscar tanto el contenido de un documento como sus metadatos asociados.

Por ello es necesario entender cada uno de los conceptos que componen la aplicación por separado para comprender la importancia del producto final del proyecto, así como el conocimiento que se requiere para lograr el alcance final.

Los conceptos a tratar son los siguientes: Bibliotecas Digitales, Metadatos (específicamente el modelo Dublin Core) y la API Lucene, la cual será la herramienta de desarrollo a utilizar para implementar el motor de búsqueda requerido.

3.1. Bibliotecas Digitales

3.1.1. Concepto

Una biblioteca digital es una colección organizada de documentos almacenados en formato digital que a su vez ofrece los servicios de búsqueda y recuperación de información, es un entorno donde se reúnen colecciones, servicios, y personal que favorece el ciclo completo de la creación, difusión, uso y preservación de los datos, para la información y el conocimiento.

Los documentos que se encuentran en una biblioteca digital pueden ser texto, imágenes, video o combinaciones de los anteriores. Idealmente se deben de almacenar y poder recuperar documentos completos, y las búsquedas se realizan sobre el contenido completo de los documentos. Es decir, si el documento es texto, la búsqueda se realiza sobre el texto completo del documento y una vez localizado el documento deseado es posible obtenerlo de manera inmediata.

Esto contrasta con los sistemas tradicionales que se concentran solamente en búsquedas basadas en el título, descripción o palabras clave. Esta sería la definición de una biblioteca electrónica, donde cada artículo es indexado por referencias pero no por su contenido, un concepto totalmente diferente a lo que se pretende lograr con este proyecto.

Existen diferentes definiciones de qué es una Biblioteca Digital, en su concepto más simple, una biblioteca digital es un espacio en donde la información es almacenada y procesada en formato digital. La definición tomada por la Digital Libraries Federation [1] podría ser, por sencillez y precisión, la más completa:

"Las Bibliotecas Digitales son organizaciones que proveen los recursos, incluyendo personal especializado, para seleccionar, estructurar, distribuir, controlar el acceso, conservar la integridad y asegurar la persistencia a través del tiempo de colecciones de trabajos digitales que estén fácil y económicamente disponibles para usarse por una comunidad definida o para un conjunto de comunidades."

La Association of Research Libraries [2] resume la mayoría de las definiciones de biblioteca digital en los siguientes elementos comunes:

- Por lo general no es un ente aislado, sino que está integrado por diversas colecciones de documentos creados y administrados por diferentes organizaciones.
- Requiere tecnologías específicas para compartir y enlazar recursos dispersos.
- Los enlaces entre diversas colecciones y servicios de información deben ser transparentes para el usuario.
- Las colecciones digitales no se restringen a sustitutos de documentos, también contienen elementos que no pueden ser representados o distribuidos en formato impreso, como el audio o video.

El objetivo principal del concepto de una biblioteca digital es el acceso universal a la información, sin limitantes de tiempo ni espacio. Altamente ligado a este objetivo están:

- Preservación a largo plazo: Las bibliotecas digitales deben estar comprometidas a preservar los materiales digitales a largo plazo.
- Acceso a largo plazo: El acceso al material debe respetarse al paso del tiempo. Tanto un documento actual como uno histórico o antiguo deberán tener las facilidades para ser consultados.

3.1.2. Estructura

Para que una colección de archivos digitales almacenados sea una biblioteca digital debe soportar todo tipo de material audio, video, imágenes y texto. Adicionalmente debe tener servicios de almacenamiento y recuperación de información. La búsqueda debe tener la posibilidad de indagar sobre el contenido de cada documento, es decir, no solo a sus datos de referencia, como autor, titulo, año, sino también al contenido interno del archivo. Obviamente, existen limitantes

en este sentido para los documentos en otros formatos diferente a texto, pero todo apunta a que en futuro podrán indexarse desde sonidos hasta el contenido de imágenes y videos. Mientras tanto, los metadatos son de vital importancia para este tipo de material digital.

Lo más común es que la información y los repositorios se encuentren de manera centralizada, pero las inmensas cantidades de colecciones digitales y los proyectos actuales de investigación sobre el área de Bibliotecas Digitales tienden a crear frameworks y estándares para que cada institución o individuo interesado diseñe su propia Librería o cargue en la red sus archivos digitales. Lo importante es que las búsquedas y las transferencias de un sistema a otro deberán ser transparentes para el usuario.

Las bibliotecas digitales proporcionan sus servicios a través de alguna red de cómputo, ya sea una red sólo de uso local o tan abierta como Internet (Ver figura 1), esto dependerá de los servicios y las restricciones que deban hacerse para el acceso a la información [3].

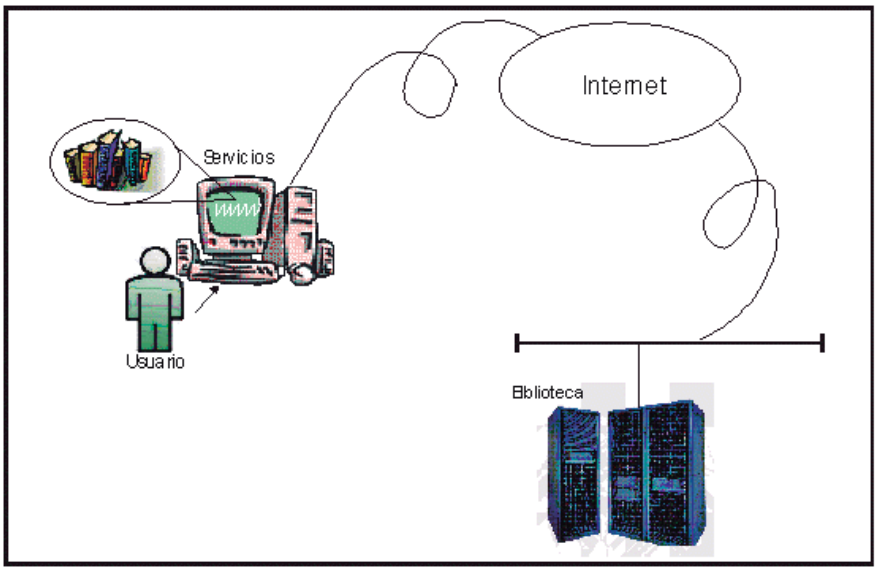


Figura 1. Las bibliotecas digitales prestan sus servicios a través de Internet o una red local

3.1.3. Ventajas y Desventajas

Las ventajas de las Bibliotecas Digitales van más allá del almacenamiento y búsqueda de documentos. También está ligada al acceso universal de la información, tanto en tiempo como en espacio, a la preservación de documentos, al acceso a documentos que eran prohibidos debido a su importancia y delicadeza física, como manuscritos antiguos o volúmenes especiales de cierta editorial.

Los beneficios de implementar bibliotecas digitales los podemos dividir para tres sectores [3]. Inicialmente, tiene beneficios nacionales o globales:

- Promueve y facilita la expansión de la cultura en una comunidad.

- Preserva en un medio no degradable las colecciones culturales y científicas generadas para el beneficio social.
- Se hace uso eficiente de los contenidos de los materiales a través de búsquedas sencillas y eficientes.
- Promueve el uso de estándares para el manejo de información digital, incluso a nivel mundial.

Los beneficios institucionales son:

- Elimina duplicidad de actividades, documentos y costos.
- Promueve nuevas áreas de investigación.
- Permite el crecimiento de colecciones sin demanda de espacio físico para almacenamiento ni para servicio.
- Prolonga la permanencia de documentos dentro de una colección disponible al público.
- Control total sobre la información.
- Reducción del costo de imprenta, para aquellas que editan sus propios libros.

Los beneficios al usuario:

- Confianza en el contenido de los documentos que se consultan.
- Acceso uniforme desde cualquier punto de la red sin desplazamiento a la biblioteca.
- Equidad en el acceso a la información, toda la comunidad autorizada tiene la misma facilidad y derecho de consultarla.
- Siempre habrá disposición de los materiales existentes que sean solicitados.

- Ahorro en el tiempo de búsqueda por la centralización aparente de los documentos y por la posibilidad de búsqueda en el contenido completo de los documentos.
- Acceso a información interrelacionada, es decir, posibilidad de enlaces hipertextuales, incluso con otros recursos de la biblioteca digital.

Pero para estas ventajas también se presentan inconvenientes, la mayoría de ellos ligados a la tecnología. Primero, y por ende más importante, no existe una estructura única de servicios para este tipo de aplicaciones ni estándares para el manejo de la información, ni políticas de acceso, así cada organización está libre de realizar su propia librería, pero al momento de integrarse con otros sistemas similares o compartir sus repositorios y bases de información con otros, se deben volver a indexar según la estructura del nuevo sistema. Segundo problema, se refiere a la infraestructura de redes y seguridad y eficiencia en el servicio de comunicaciones, que depende tanto del lugar donde se encuentren los repositorios, como el lugar de donde el usuario solicita acceso a la información. Tercer problema, pero que a la vez no lo es debido a la actual tecnología, se debe a la capacidad de almacenamiento masivo, ya que a mayor cantidad de información digital requieren dispositivos con mayor capacidad de almacenamiento y por ende más eficientes y más costosos.

Una desventaja complicada de tratar, independiente de la tecnología, se refiere a los derechos de propiedad del material digital. *El Web es el medio en donde la violación de la propiedad intelectual es más común* [4]. Si cualquier persona en cualquier lugar tiene acceso a cualquier contenido digital, entonces no se podrán garantizar ni los derechos de autor ni el debido uso del material contenido en los repositorios.

6 ; (;) 5 4 /=1 (((- 9 3 /1%

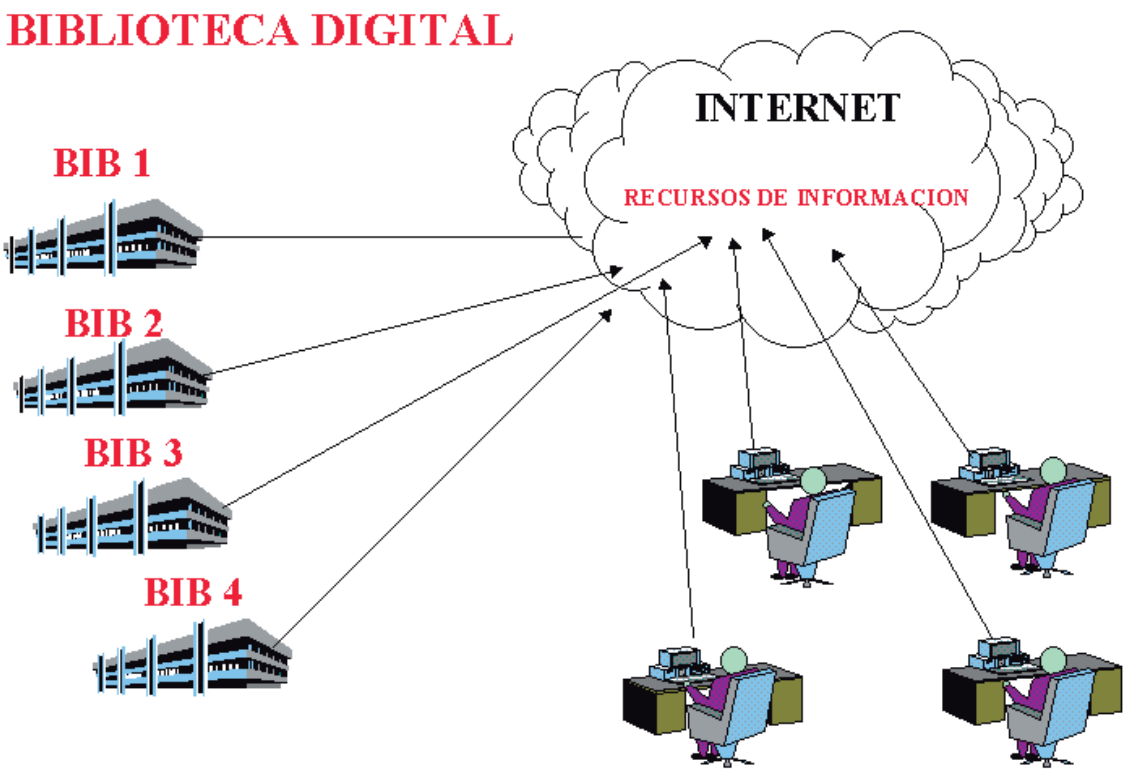


Figura 2. Concepto moderno de una biblioteca digital donde el acceso es centralizado pero el material realmente se encuentra descentralizado.

Todo con el fin de hacer más fácil la búsqueda de documentos. Dean Marcum habla de los usuarios y sus deseos de usar los recursos digitales y no de ir necesariamente a la biblioteca tradicional [7].

Y aunque, si bien, la calidad de una biblioteca se encuentra en su contenido y forma de organización y búsqueda, "la facilidad en el acceso a la información va a ser siempre un condicionante para su uso y la selección de información digital (para tenerla almacenada en servidores locales o tenerla enlazada en el catálogo o en una fuente local de recursos electrónicos) va a continuar siendo una función

de las bibliotecas si quieren así seguir cumpliendo su función de ayudar al usuario a encontrar información" [8].

3.2. Metadatos

3.2.1. Concepto

El concepto de metadatos tiene múltiples definiciones, “informaciones sobre datos” [9], “datos sobre informaciones” [10], “informaciones sobre informaciones” [11], pero al final todas concluyen con una de las definiciones más primitivas de dicho término: “dato sobre los datos” [12]. Es decir, es información que caracteriza los datos, visto desde el punto de vista de Ingeniería de Software, serían como los atributos que caracterizan un objeto, en este caso, un objeto que contiene información.

Se caracterizan porque permiten tener una descripción estandarizada de los diferentes conjuntos de datos que están presentes en el sistema. Son una herramienta de gran importancia en la gestión de datos e información porque facilitan la búsqueda, recuperación e integración de datos provenientes de distintas fuentes. Nacen de la necesidad de recuperar la información electrónica tan dispersa. Los metadatos tratan, principalmente, de describir el contenido y la localización del objeto de la información en Internet.

Su principal uso se centra en las bibliotecas tanto de contenido textual como multimedia, pues actualmente la tecnología solo permite categorizar los materiales digitales por sus propiedades externas, es decir, por sus propiedades sintácticas ya que aún no se puede indexar un video o un archivo de audio por su autocontenido como tal, sino sus propiedades como tamaño, duración, autor, etc.

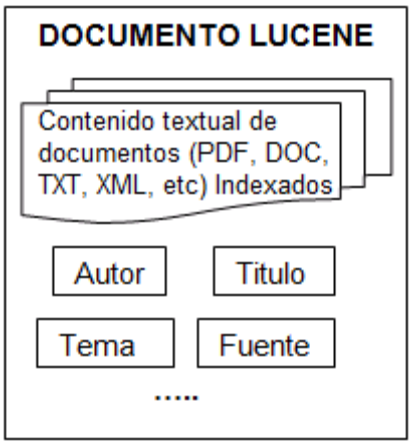
		L
4-	2	 7 7 % 7 %
,	2)	 2

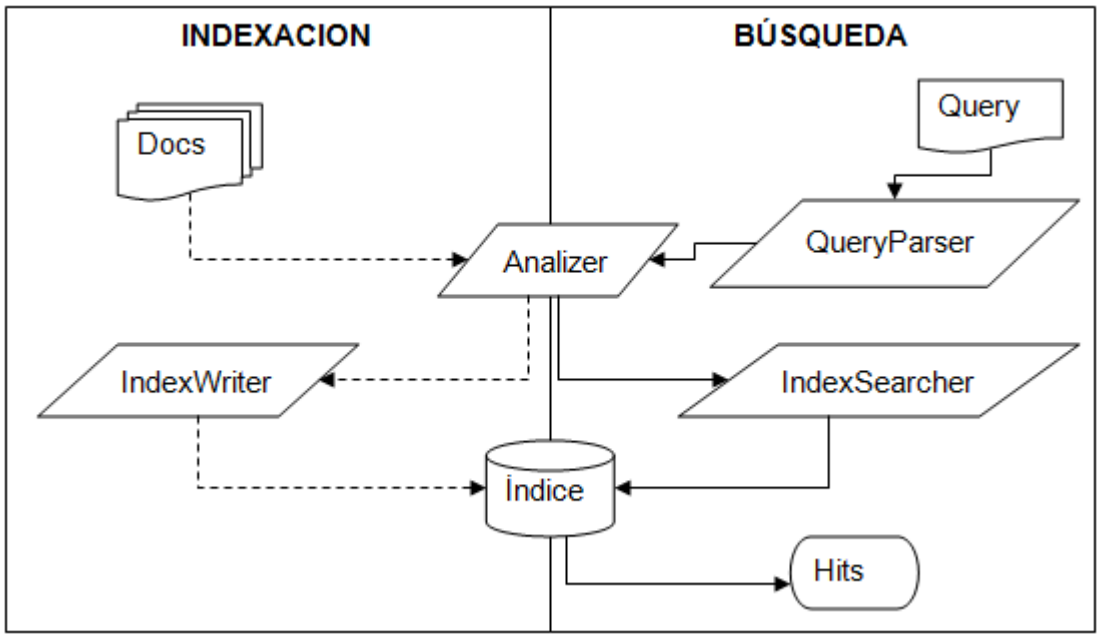
(

/0=1

2

;


DOCUMENTO LUCENE
Contenido textual de documentos (PDF, DOC, TXT, XML, etc) Indexados
Autor
Titulo
Tema
Fuente
.....


INDEXACION
BÚSQUEDA
Docs
Query
Analizer
QueryParser
IndexWriter
IndexSearcher
Índice
Hits

$$\cos\theta = \frac{\mathbf{v_1} \cdot \mathbf{v_2}}{\|\mathbf{v_1}\| \, \|\mathbf{v_2}\|}$$

; [illegible] [illegible] [illegible]

$$score(q,d) = coord(q,d) \cdot queryNorm(q) \cdot \sum_{t\ en\ q} \left(tf(t\ en\ d) \cdot idf(t)^2 \cdot t.getBoost \cdot norm(t)\right)$$

$tf(t\ en\ d)$

$$tf(t\ en\ d) = frecuencia^{1/2}$$

- $idf(t)$

$$idf(t) = 1 + log\left(\frac{numDocs}{docFreq+1}\right)$$

- $coord(q,d)$

- $queryNorm(q)$

$$queryNorm(q) = \frac{1}{sumOfSquaredWeigths^{1/2}}$$

$$sumOfSquareWeights = q.getBoost()^2 \cdot \sum_{t \ en \ q} \left(idf(t) \cdot t.getBoost()\right)^2$$

- $t.getBoost()$

$norm(t)$ [illegible]

[illegible]

[illegible]

$$norm(t,d) = doc.getBoost() \cdot lengthNorm(Field) \cdot \prod_{Campo\ f\ en\ d\ llamado\ t} f.getBoost()$$

[illegible]

[illegible]

[illegible]

/3=1

4 - - ; ; R;
, G" ; " /3 1 , G " ; " +%
7
(

/ ( ) 0
; 6 %
J 2 2 - %
6 ; ; %
I 4 %
L G 2 %
; , 2 2 G4 + %
2 6 Q, ; 4 %
2 ; " ( ) %
4L 6 " ; 6 G I %
, 46 4 %
/ 0
6 H

; □2 , ; □

□G□; □

□

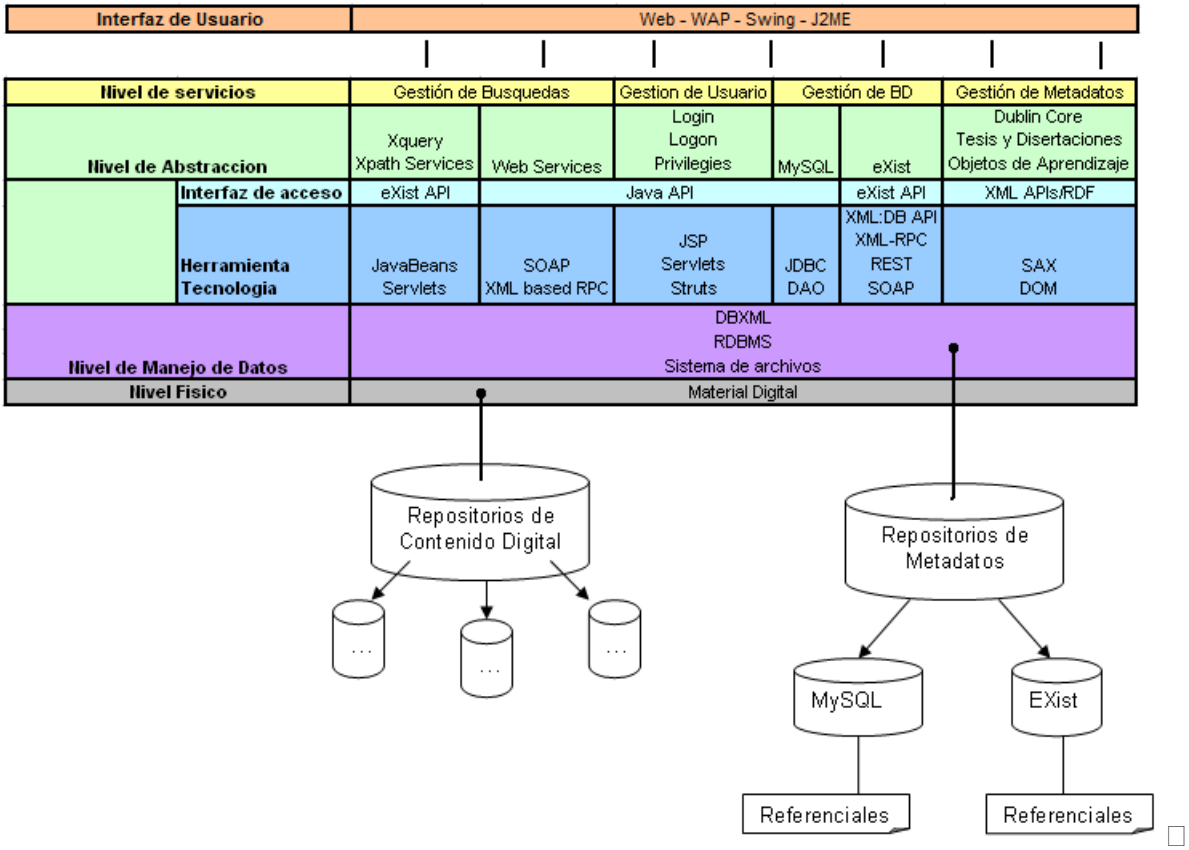
6□□□□□ □ □□□□ □□□□□□□□□□

□

L □□□□□□□□□□

6

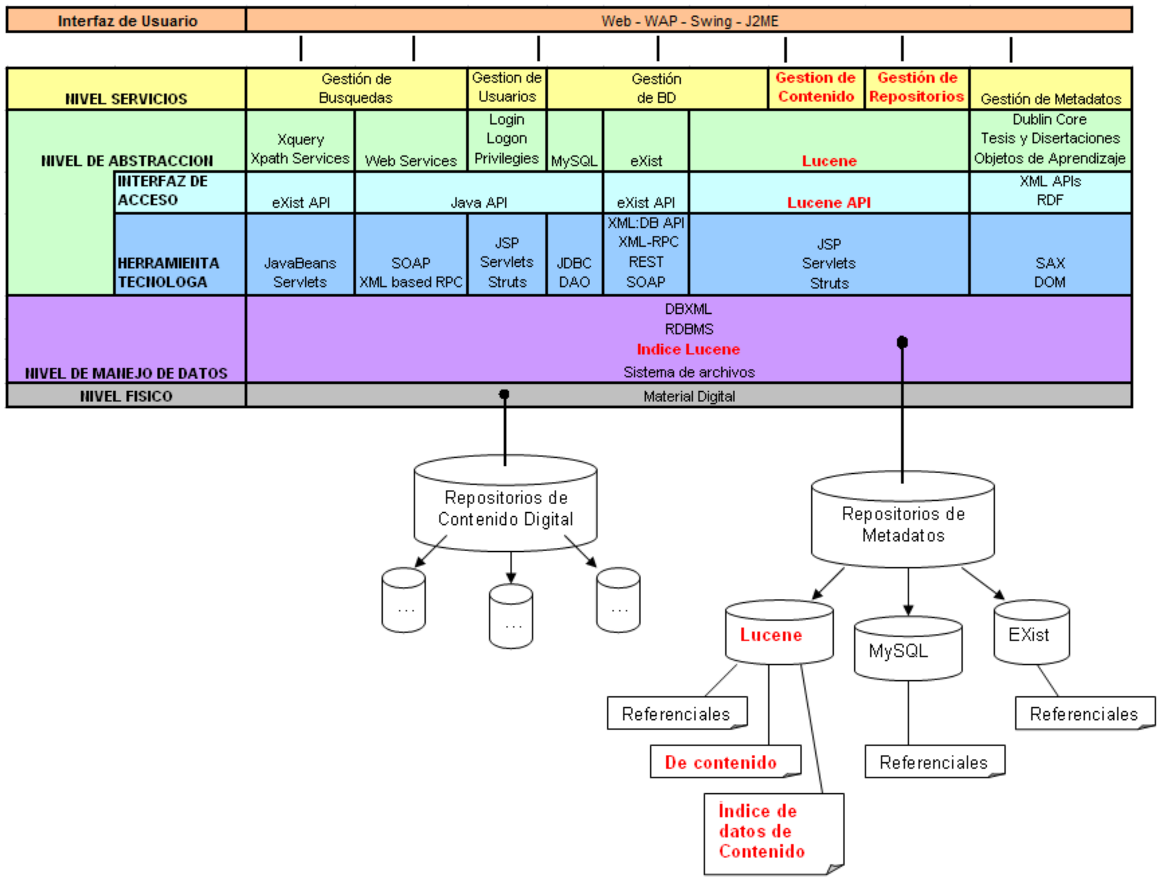
□□ G□□□) □□, □□□□□□□□□□□□7 □□□□□□□□□□□□

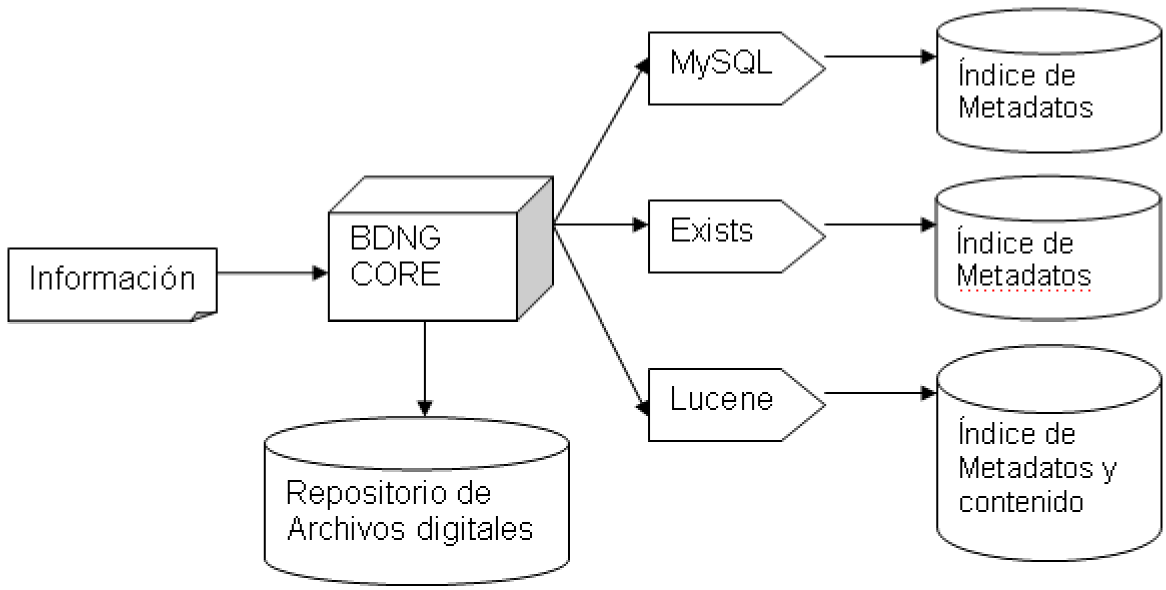

Interfaz de Usuario
Web - WAP - Swing - J2ME
Nivel de servicios
Gestión de Busquedas
Gestion de Usuario
Gestión de BD
Gestión de Metadatos
Nivel de Abstraccion
Xquery
Xpath Services
Web Services
Login
Logon
Privilegies
MySQL
eXist
Dublin Core
Tesis y Disertaciones
Objetos de Aprendizaje
Interfaz de acceso
eXist API
Java API
eXist API
XML APIs/RDF
Herramienta
Tecnologia
JavaBeans
Servlets
SOAP
XML based RPC
JSP
Servlets
Struts
JDBC
DAO
XML:DB API
XML-RPC
REST
SOAP
SAX
DOM
Nivel de Manejo de Datos
DBXML
RDBMS
Sistema de archivos
Nivel Fisico
Material Digital
Repositorios de
Contenido Digital
Repositorios de
Metadatos
MySQL
EXist
Referenciales
Referenciales

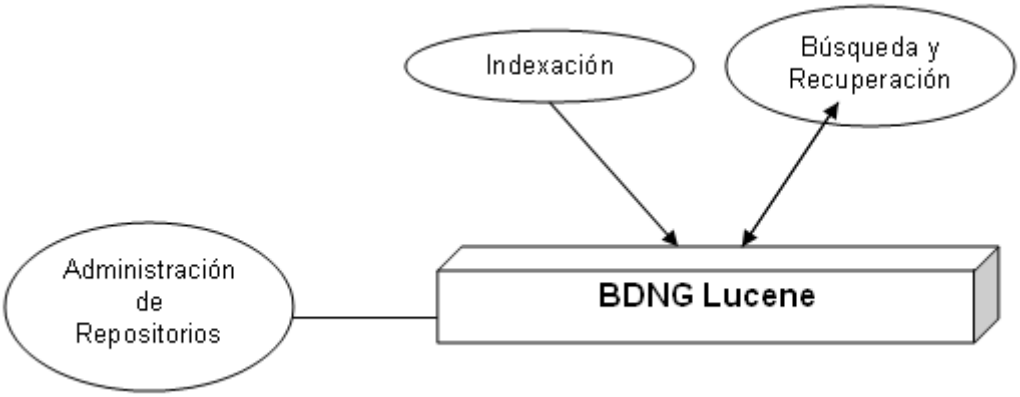
7 GI

GI

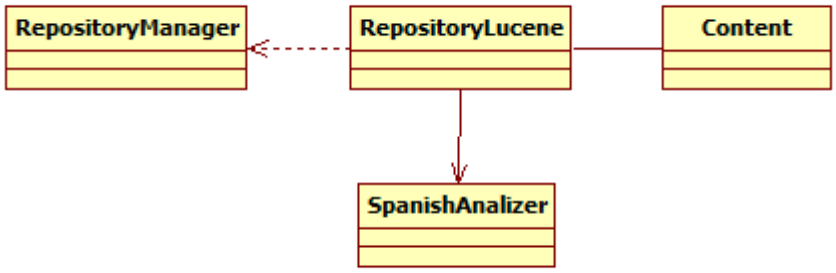
; [illegible] [illegible] [illegible]


Interfaz de Usuario
Web - WAP - Swing - J2ME
NIVEL SERVICIOS
Gestión de Busquedas
Gestion de Usuarios
Gestión de BD
Gestion de Contenido
Gestión de Repositorios
Gestión de Metadatos
NIVEL DE ABSTRACCION
Xquery
Xpath Services
Web Services
Login
Logon
Privilegies
MySQL
eXist
Lucene
Dublin Core
Tesis y Disertaciones
Objetos de Aprendizaje
INTERFAZ DE ACCESO
eXist API
Java API
eXist API
Lucene API
XML APIs
RDF
HERRAMIENTA TECNOLOGA
JavaBeans
Servlets
SOAP
XML based RPC
JSP
Servlets
Struts
JDBC
DAO
XML:DB API
XML-RPC
REST
SOAP
JSP
Servlets
Struts
SAX
DOM
NIVEL DE MANEJO DE DATOS
DBXML
RDBMS
Indice Lucene
Sistema de archivos
NIVEL FISICO
Material Digital
Repositorios de Contenido Digital
Repositorios de Metadatos
Lucene
MySQL
EXist
Referenciales
De contenido
Referenciales
Referenciales
Indice de datos de Contenido


Información
BDNG CORE
Repositorio de Archivos digitales
MySQL
Exists
Lucene
Índice de Metadatos
Índice de Metadatos
Índice de Metadatos y contenido


Indexación
Búsqueda y
Recuperación
Administración
de
Repositorios
BDNG Lucene

7



GI

4

4

GI]

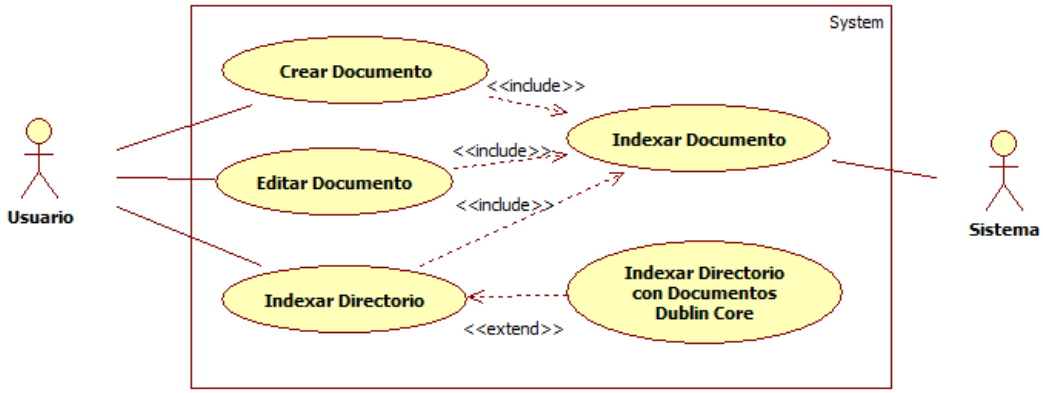
2

7 %

5

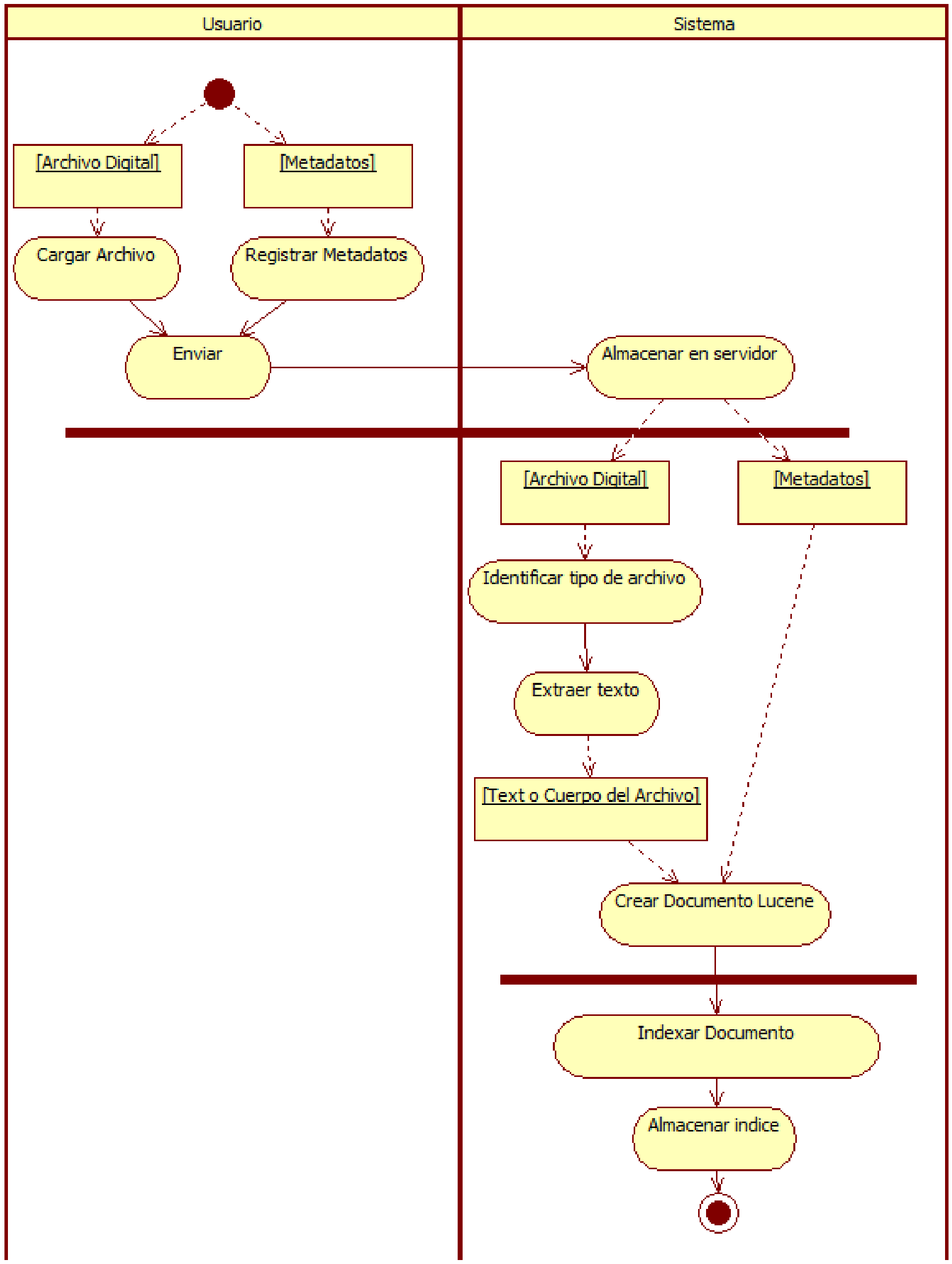
7

7 1)


System
Crear Documento
<<include>>
Indexar Documento
<<include>>
Editar Documento
<<include>>
Usuario
Sistema
Indexar Directorio
Indexar Directorio con Documentos Dublin Core
<<extend>>

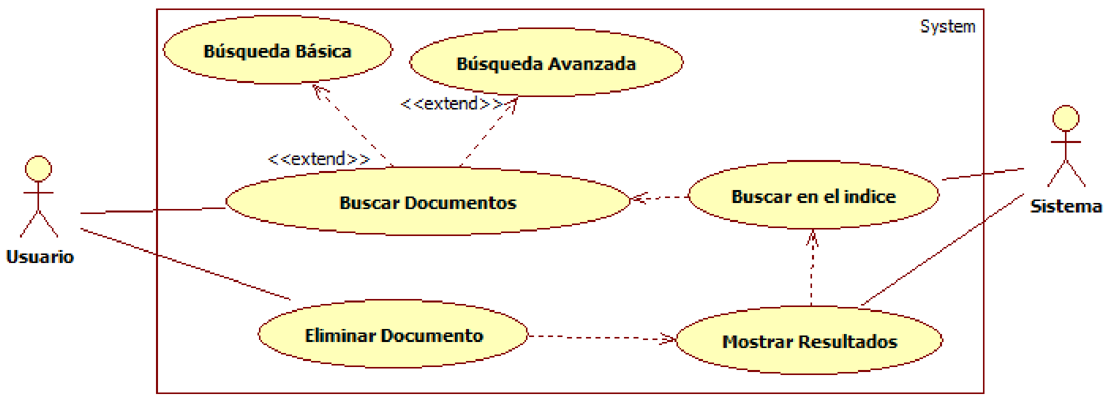
A

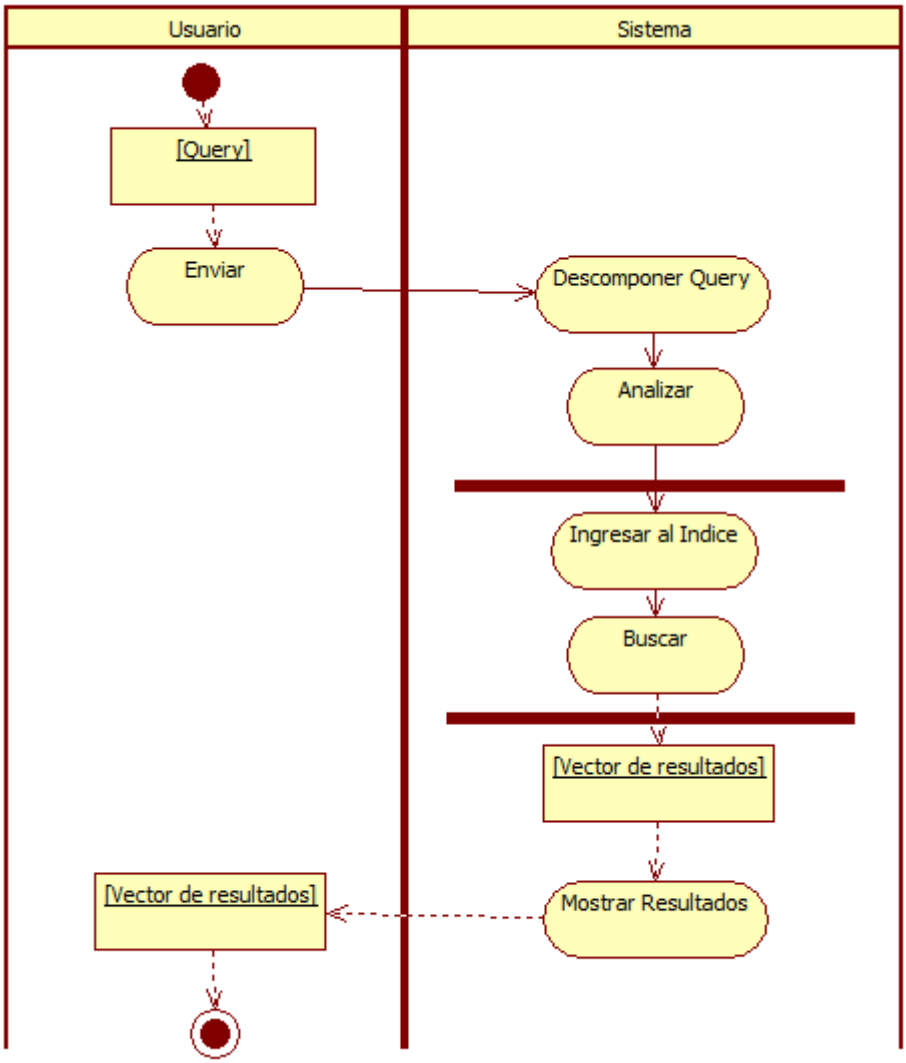
N) 5 GI)


Usuario
Sistema
[Archivo Digital]
[Metadatos]
Cargar Archivo
Registrar Metadatos
Enviar
Almacenar en servidor
[Archivo Digital]
[Metadatos]
Identificar tipo de archivo
Extraer texto
[Text o Cuerpo del Archivo]
Crear Documento Lucene
Indexar Documento
Almacenar indice

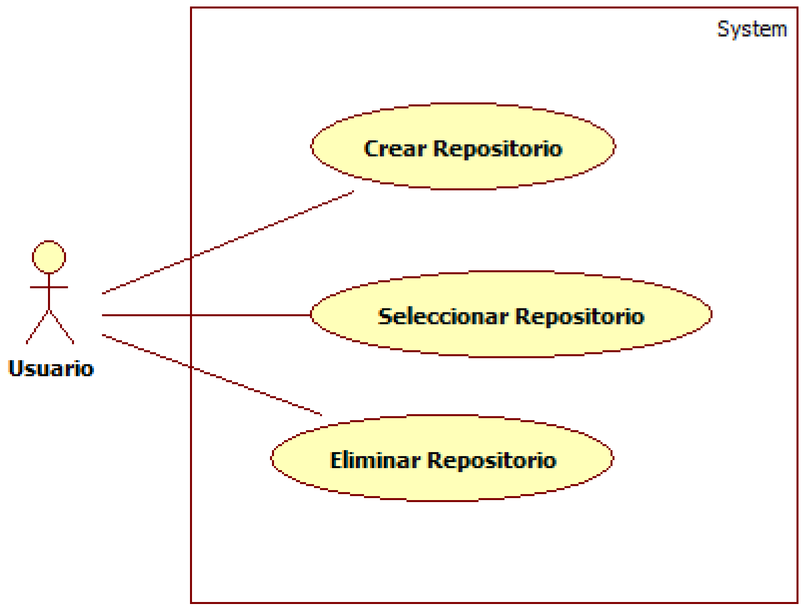
[illegible]

7 [illegible] B [illegible]


System
Búsqueda Básica
Búsqueda Avanzada
<<extend>>
<<extend>>
Buscar Documentos
Buscar en el indice
Eliminar Documento
Mostrar Resultados
Usuario
Sistema


Usuario
Sistema
[Query]
Enviar
Descomponer Query
Analizar
Ingresar al Indice
Buscar
[Vector de resultados]
[Vector de resultados]
Mostrar Resultados

[illegible]


System
Crear Repositorio
Seleccionar Repositorio
Eliminar Repositorio
Usuario

[illegible]

[illegible]

[illegible]

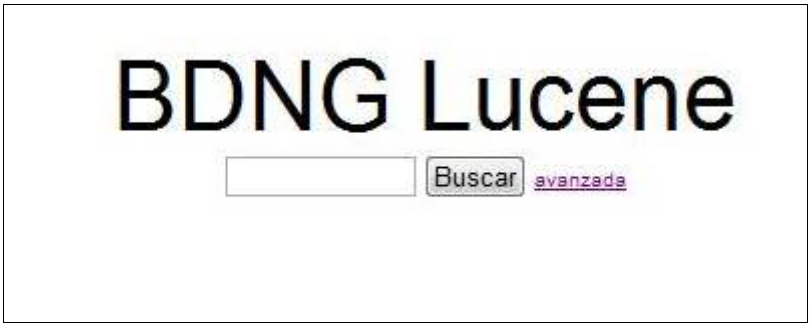
[illegible]

+ 5

3 +

7 B

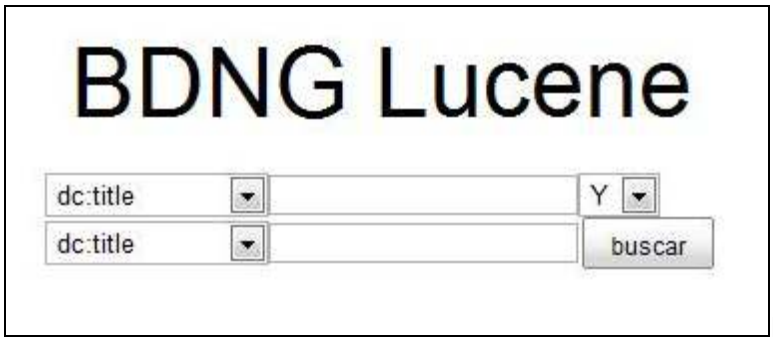
+



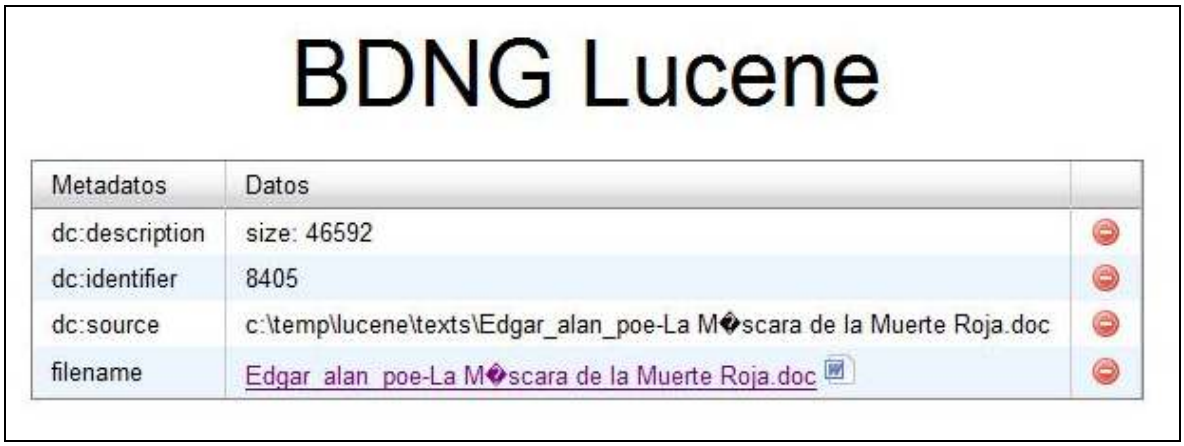
4 -

+

)

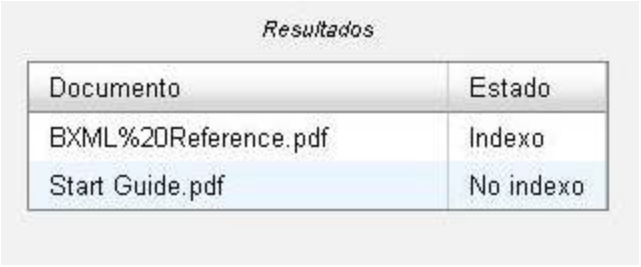

BDNG Lucene
dc:title
Y
dc:title
buscar


Resultados
Id
Título
Documento
8405 _ Edgar_alan_poe-La M�scara de la Muerte Roja.doc
49614 _ ejemplo1.xml
38369 _ 5d_Miguelez_122151003_esp.pdf
92719 _ descargar.doc
54789 _ Efesios.doc
66584 _ EL EVANGELIO DE LA NATIVIDAD DE MARIA.doc
75077 _ 1Corin.doc
79965 _ CONDE DE MONTECRISTO, EL - Alejandro Dumas.doc
85542 _ Jorge Luis Borges - Ficciones.pdf


BDNG Lucene
Metadatos
Datos
dc:description
size: 46592
dc:identifier
8405
dc:source
c:\temp\lucene\texts\Edgar_alan_poe-La M�scara de la Muerte Roja.doc
filename
Edgar_alan_poe-La M�scara de la Muerte Roja.doc

7


Metadato
Valor
dc:title
titulo
dc:language
español
dc:format
alguno
Añadir fila
C:\Users\JSILVAZU\Des
Browse...
Registro


Resultados
Documento
Estado
BXML%20Reference.pdf
Indexo
Start Guide.pdf
No indexo


Seleccione el Repositorio a utilizar
Repositorio
aaa
crisa
index
iuiuiu
Adicionar

□

□□□

2 [illegible] 3FF [illegible] %[illegible]

2 [illegible] M' [illegible]

[illegible] Gl [illegible] G [illegible] 9 [illegible]

7. CONCLUSIONES

Las bibliotecas digitales cada vez más se ubican dentro de un marco tecnológico promisorio para el manejo de la información, han evolucionado su concepto enfocadas en el servicio al cliente, pues hoy día no es suficiente que se almacenen los metadatos y los sistemas retornen la ubicación del archivo coincidente, sino que el usuario exige resultados más precisos en cuanto al contenido del material digital.

Se desarrolló el núcleo de una biblioteca digital moderna, la cual indexa tanto sus datos referenciales o metadatos como su contenido. Adicionalmente, el sistema permite satisfacer las necesidades de búsqueda de los usuarios, quienes usualmente basan sus búsquedas en datos referenciales o palabras clave, recuperando documentos según su contenido interno.

Durante el proceso fue una tarea fundamental la investigación acerca de las herramientas que permitieran lograr el alcance definido al inicio del proyecto. Lucene cumple con todos los requisitos necesarios para la implementación, además de sus potentes opciones de indexación y búsqueda, se puede agregar su sencillez de implementación y completa documentación como características a favor en comparación de otras herramientas.

Dado que el BDNG Core aún se encuentra en etapa de laboratorio y sujeto a cambios, la aplicación BDNG Lucene funciona independientemente, aunque bajo la estructura actual de BDNG Core. Lo anterior deja la puerta abierta para que se integre a dicho sistema o se desarrolle como un sistema independiente al servicio de la comunidad universitaria.

8. BIBLIOGRAFIA

[1] Digital Library Federation. "*A Working Definition of Digital Library".* Octubre 1998.- http://www.clir.org/diglib

[2] Association of Research Libraries. http://www.arl.org.

[3] LOPEZ GUZMAN, Clara. "*Specialized digital libraries development model*". Marzo 2000. -www.bibliodgsca.unam.mx

[4] GLADNEY, Henry. "*Digital Dilemma: Intellectual Property*". Diciembre 1999.- http://www.dlib.org/dlib/december99/12gladney.html

[5] CORRAL, Ana. ORDAS, Ana. "*Bibliotecas digitales*". http://www.absysnet.com/tema/tema41.html

[6] El Dorado, Mexico, Biblioteca Virtual Iberoamericana y del Caribe / Presentación. http://eldorado.ucol.mx/menudecontenidos.php?clave=1&pagina=InicioF.htm

[7] MARCUM, Deane. "*Requeriments for the Future Digital Library".* The Joournal of Academis Librarianship, vol. 29, 2003, pág 277

[8] ANGLADA I DE FERRER, Lluís Ma. *"Biblioteca digital ¿mejor, peor o solo distinto?"* Anales de Documentación, nº 3, 2000, pág. 27

[9] SHELDON, Tom. Linktionary. Entrada «Metadata». 2001. http://www.linktionary.com/m/metadata.html

[10] STEINACKER, A. GHAVAM, A. STEINMETZ, R. "M*etadata Standards for Web-Based Resources"*. *IEEE MultiMedia,* enero-marzo 2001. http://www.dsc.ufcg.edu.br/~garcia/cursos/TEICOPIN/metadataWE.pdf

[11] SWICK, Ralph. (W3C), "*Metadata Activity Statement*". 2002. http://www.w3.org/Metadata/Activity.html

[12] Originariamente este concepto fue ideado por Jack Myers en la década de los años sesenta para describir datos. MILLER, Paul. "*Metadata for the masses*". http://www.ariadne.ac.uk/issue5/metadata-masses/

[13] GILL, Tony. BACA, Murtha. GILLILAND-SWETLAND, Anne. "*Introducción a Los Metadatos: Caminos a la Información Digital*", Mayo 1999, Getty Publications. http://portal.acm.org/citation.cfm?id=553735&coll=GUIDE&dl=GUIDE&CFID=4804961&CFTOKEN=65363249

[14] XU, Amanda. "*Accessing information on Internet*". OCLC, 1996.

[15] DCMI – Dublin Core Metadata Initiative. http://www.dublincore.org

[16] DCMI – Expressing Qualified Dublin Core in RDF / XML. http://dublincore.org/documents/dcq-rdf-xml/

[17] HERMA, Iván. (W3C) "*Semantic Web Activity*". http://www.w3.org/2001/sw/

[18] Lucene. The Apache Open-Source Search Project. http://lucene.apache.org/

[19] GOSPODNETIC, Otis. HATCHER, Eric. "*Lucene In Action*", Manning Publications, 2005. ISBN 1-932394-28-1

[20] DELOS Network of Excellence on Digital Libraries. http://www.delos.info/

[21] Greenstone Digital Library Software. http://www.greenstone.org/

[22] DSPACE, an open-source solution for accesing, managing and preserving scholarly works. http://www.dspace.org/

[23] LAGOZE, Carl. VAN DE SOMPEL, Herbert. NELSON, Michael. WARNER, Simeon. "*The Open Archives Initiative Protocol for Metadata Harvesting*" http://www.openarchives.org/OAI/openarchivesprotocol.html

[24] El Dorado, México, Biblioteca Virtual Iberoamericana y del Caribe. http://eldorado.ucol.mx/

[25] El Dorado, México, Biblioteca Virtual Iberoamericana y del Caribe / Antecedentes / ¿Cómo se estructuró la biblioteca digital?, http://eldorado.ucol.mx/menudecontenidos.php?clave=1&pagina=InicioF.htm

[26] SISBIB – Sistema de Bibliotecas (Universidad Nacional Mayor de San Marcos). http://sisbib.unmsm.edu.pe/

[27] RENATA – Red Nacional Académica de Tecnología Avanzadahttp://www.renata.edu.co/

[28] RENATA – "*Creación de la Biblioteca Digital Colombiana*". Agosto 2008. http://www.renata.edu.co/informacion-de-proyectos/creacion-de-la-biblioteca-digital-colombiana.html

[29] Metadata Encoding and Transmission Standard. http://www.loc.gov/standards/mets/

[30] The Federal Geographic Data Committee. http://www.fgdc.gov

[31] Contents Standards for Digital Geospatial Metadata http://geology.usgs.gov/tools/metadata/standard/metadata.html

[32] ISO 19115:2003 Geographic Information Metadata http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=26020

[33] The library of Congress – Enconded Archival Description Version 2002. http://www.loc.gov/ead

[34] SPERBERG-MCQUEEN, C. M.; BURNARD, Lou. “Text Encoding For Interchange”. http://www.tei-c.org/Lite/teiu5_sp.html#ID3

[35] Visual Resources Association - The International Association of Image Media Professionals. http://www.vraweb.org/

[36] World Wide Web Consortium. http://www.w3c.org/

[37] HERMAN, Iván; SWICK, Ralph; BRICKLEY, Dan. “*Resource Definition Framework*”. (W3C) http://www.w3.org/RDF

[38] BRICKLEY, Dan; GUHA, R.V.; MCBRIDE, Brian. “Vocabulary Description Language: 1.0. RDF Schema”. (W3C). http://www.w3.org/TR/rdf-schema

[39] Online Computar Library Center. http://www.oclc.org/

[40] La información sobre estos proyectos de OCLC puede encontrarse en: http://www.oclc.org/research/projects/default.htm

[41] THOMSON, R.; SHAFER, K; VIZINE-GOETZ, D. “*Evaluating Dewey Concepts as a Knowledge Base for Automatic Subject Assignement*”. http://portal.acm.org/citation.cfm?id=263690.263790&coll=portal&dl=ACM&CFID=15151515&CFTOKEN=6184618

[42] National Document and Information Service (NDIS) Project http://www.nla.gov.au/policy/annrep95/stl.html#ndis

[43] MARC Standards. http://www.loc.gov/marc/marcspa.html

[44] CAPLAN, Priscila; GUENTHER, Rebecca. “Metadata for Internet Resources: The Dublín Core Metadata Elements Set and its Mapping to USMARC”.

[45] UKLON Software Tools. http://www.ukoln.ac.uk/metadata/software-tools/

[46] NUTCH, the open-source Web-Search. http://lucene.apache.org/nutch/

[47] REGAIN. http://regain.sourceforge.net/

[48] Documentation API Lucene. Simility Class. http://hudson.zones.apache.org/hudson/job/Lucene-trunk/javadoc//index.html?org/apache/lucene/search/Similarity.html

[49] MOLE – Text Analysis Group. Vector Space Model. http://isp.imm.dtu.dk/thor/projects/multimedia/textmining/node5.html

[50] Documentation API Lucene. IndexWriter Class. http://hudson.zones.apache.org/hudson/job/Lucene-trunk/javadoc//org/apache/lucene/index/IndexWriter.html

[51] Modelos de recuperación de información. http://modelosderecuperacion.googlepages.com/index.htm

MANUAL DE USUARIO

Sistema BDGN Lucene

CRISTIAN RÍOS VILLADA

JOHN JAIRO SILVA ZULUAGA

Medellín

Universidad Eafit

2008

TABLA DE CONTENIDO

INTRODUCCIÓN

BDNG Lucene es una aplicación web basada en el núcleo BDNG (Biblioteca Digital Nueva Generación) de la universidad EAFIT con el fin de crear una librería virtual donde los usuarios puedan ingresar archivos con sus respectivos metadatos y las opciones de Buscar, Actualizar y Eliminar dichos documentos.

Los procesos de almacenamiento e indexación de la información son realizados con la ayuda de la librería de código abierto Lucene, la cual sirve de herramienta para convertir la aplicación en un potente y eficaz motor de búsqueda.

Este documento se encarga de describir funcionalmente el sistema asi como el proceso de instalación. Los detalles de diseño y estructura se encuentran en el manual técnico.

REQUERIMIENTOS PARA EJECUCIÓN

HARDWARE

Mínimo:

- **Procesador:** 700 MHz
- **RAM:** 512 MB

Recomendado:

- **Procesador:** 1.8 GHz
- **RAM:** 1 GB

SOFTWARE

- Apache Tomcat 6 (http://tomcat.apache.org/download-60.cgi)
- Java 6. (http://java.sun.com/javase/6/)
- **Browser:** Probado en Internet Explorer, Mozilla Firefox o Google Chorme.

PASOS PARA EJECUTAR LA APLICACIÓN

1. Descargar e instalar el Apache Tomcat 6.
2. En el CD del proyecto BDNG Lucene, se encuentra el archivo BDNGlucene.war
3. Copiar dicho archivo en la carpeta *tomcat*/webapps la cual se encuentra en la ruta donde fue instalado el tomcat.

   O abrir el Tomcat Manager (http://127.0.0.1:8080/manager/html) y cargar el dicho archivo en la sección Desplegar - Archivo WAR a desplegar. Seleccionar la ruta del archivo y oprimir el botón *desplegar*.

   En ambos casos el proyecto debe aparecer en la sección de Aplicaciones del Tomcat Manager.
4. Luego se debe abrir el browser preferido por el usuario y digitar alguna de las siguientes rutas:

   - http:// localhost:8080/luceneweb/
   - http://127.0.0.1:8080/luceneweb/

   Ambas rutas corresponden a la configuración por defecto del servidor local de Tomcat, si dicha configuración, tanto la ruta como el puerto fueron cambiados, entonces el usuario deberá digitar dicha ruta y el nombre del proyecto */BDNGlucene*

MODULOS FUNCIONALES

El sistema posee 2 interfaces, una de búsqueda y otra de administración. La primera es un modo de solo consulta por medio de búsqueda básica y búsqueda avanzada; en la segunda el usuario puede agregar documentos, indexar directorios completos, indexar directorios de documentos xml Dublin Core y administrar los repositorios.

Por defecto, al digitar la url de ingreso (mencionada en paso 4 de la sección anterior), el sistema despliega inmediatamente el modo de búsqueda.

MODO DE BÚSQUEDA

Se puede realizar una búsqueda básica directamente o se puede ingresar a realizar una búsqueda avanzada.

Pero antes de entrar en detalle es conveniente realizar una especificación del lenguaje de búsqueda que soporta el API Lucene.

Lenguaje de Búsqueda

Operadores.

Dentro de los 2 tipos de búsqueda, el usuario puede ingresar términos y operadores lógicos en sus campos de búsqueda:

Un término individual es una sola palabra, como “test” o “hello”.

Una frase es un grupo de palabras rodeado por comillas, como “hello dolly”.

Puede buscar en cualquier campo escribiendo el nombre del campo seguido de dos puntos ":" y, después, el término que está buscando.

Como ejemplo, supongamos que un índice Lucene contiene dos campos, título y texto, y que texto es el campo predeterminado. Si quiere encontrar el documento titulado “The Right Way”, que contiene el texto “don't go this way”, puede escribir:

- title:"The Right Way" AND text:go
- title:"Do it right" AND right. Como el texto es el campo predeterminado, el indicador de campo no es necesario.

Sin embargo, para evitarle este tipo de complicaciones al usuario, la busqueda avanzada implícitamente se encarga de construir dicho query, y solo basta con seleccionar los campos en los que se debe buscar y los criterios de búsqueda.

Nota: El campo sólo es válido para el término al que precede directamente, de forma que

- title:Do it right

Sólo encontrará "Do" en el campo del título. Encontrará "it" y "right" en el campo predeterminado (en este caso, el campo de texto).

Modificadores de texto

Lucene es compatible con la modificación de términos de consulta para proporcionar un amplio rango de opciones de búsqueda.

- Búsquedas comodín

Lucene es compatible con búsquedas de caracteres comodín individuales y múltiples.

Para realizar una búsqueda comodín de un solo carácter, use el símbolo "?".

Para realizar una búsqueda comodín de varios caracteres, use el símbolo "*".

La búsqueda comodín de un solo carácter busca términos que coincidan con el carácter individual que se ha sustituido. Por ejemplo, para buscar "text" o "test" puede usar la búsqueda: te?t

La búsqueda comodín de varios caracteres busca 0 ó más caracteres. Por ejemplo, para buscar *test*, *tests* o *tester*, puede usar la búsqueda: test*

También puede usar las búsquedas comodín en medio de un término. te*t

Nota: No puede usar un símbolo * ni ? como primer carácter de una búsqueda.

- Búsquedas difusas

Lucene es compatible con búsquedas difusas basadas en el algoritmo de la Distancia Levenshtein o en el algoritmo de Distancia de edición. Para realizar una búsqueda difusa, use el símbolo de la tilde, "~", al final de una

búsqueda de una sola palabra. Por ejemplo, para buscar un término que se deletree de forma parecida a "roam" use la búsqueda difusa: roam~

Esta búsqueda encontrará términos como “foam” y “roams”.

- Operadores booleanos

Los operadores booleanos permiten que los términos se combinen mediante operadores lógicos. Lucene acepta AND, "+", OR, NOT y "-" como operadores booleanos (Nota: los operadores booleanos deben estar COMPLETAMENTE EN MAYÚSCULAS).

Para mas información acerca del tipo de búsquedas que se pueden realizar por medio de Lucene leer el siguiente documento:

- Sintaxis del Analizador de búsquedas de Lucene:

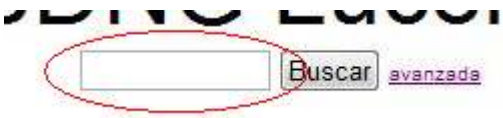
  http://www.ehu.es/p200-content/es/contenidos/informacion/ayuda_busquedas/es_def/adjuntos/Lucene-Overview.pdf

  (De la página de la Universidad del Pais Vasco, pero no tiene autor ni año de publicación, sin embargo es un material didáctico que comprende las búsquedas que se pueden realizar por medio de Lucene).

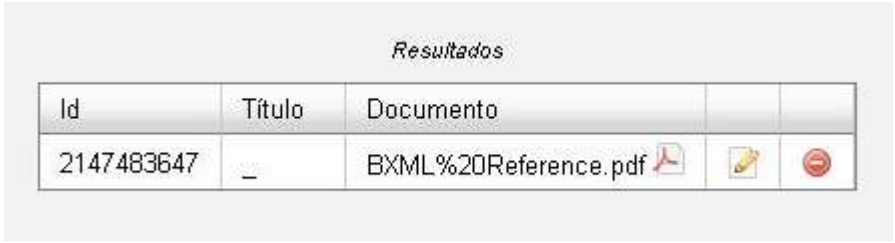
Búsqueda básica

Esta búsqueda consiste en ingresar los parámetros de búsqueda y el sistema se encargará de retornar los documentos coincidentes sin importar en que metadato o sección del documento se encuentran las palabras a buscar.

Para esta búsqueda se debe ingresar la(s) palabra(s) a buscar y oprimir el botón buscar.



El sistema retornará los resultados que coincidan con esta búsqueda y los campos de Id, Titulo y Documento.

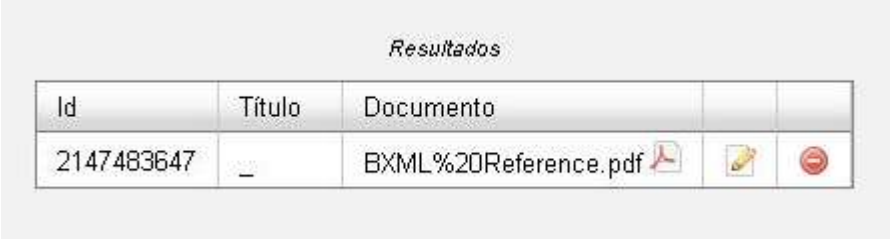


Búsqueda avanzada

Para ingresar a esta búsqueda se debe ingresar seleccionar el link de búsqueda avanzada, al lado del botón Buscar.

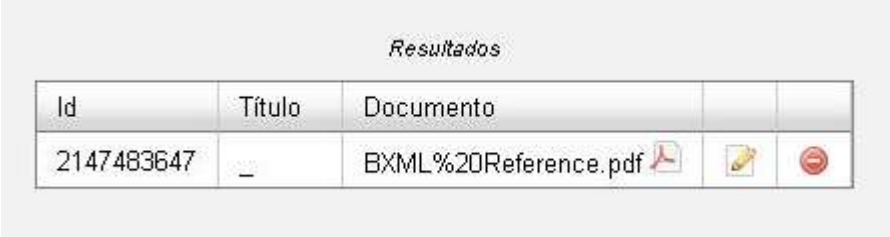
Para esta búsqueda se pueden seleccionar los campos específicos y las palabras que se desean buscar, y los operadores lógicos entre dichos campos para hacer más precisa la búsqueda.

Al oprimir el botón buscar se muestran los resultados según los criterios de búsqueda.

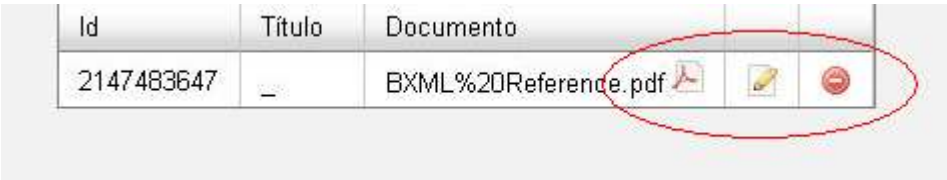


Ver resultados

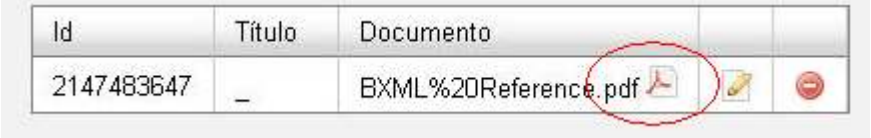
Sin importar el tipo de búsqueda, los resultados solo mostrarán algunos de los datos básicos asociados al documento.



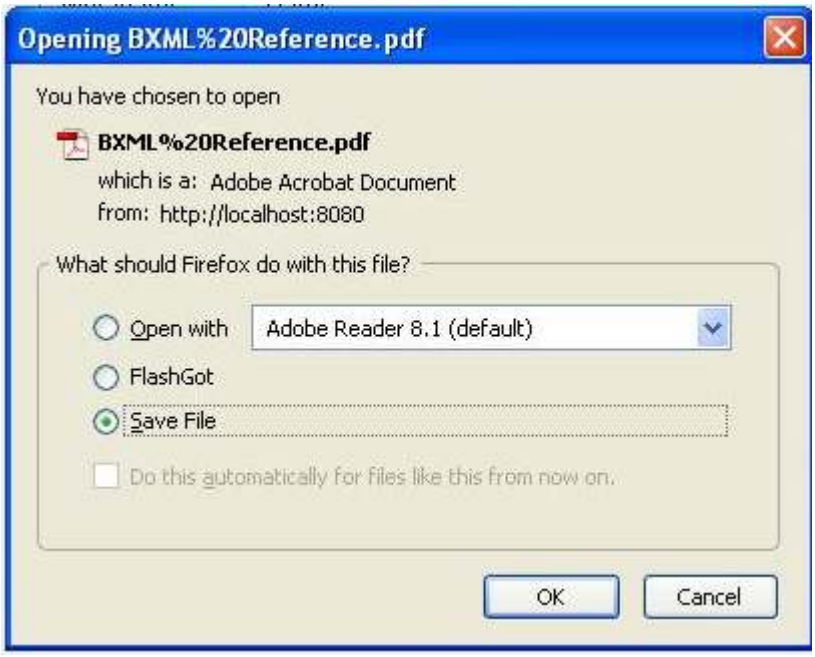
Cada registro encontrado tendrá la posibilidad de ver el detalle y descargar documento.



Para descargar el documento se debe dar click sobre el icono del documento.

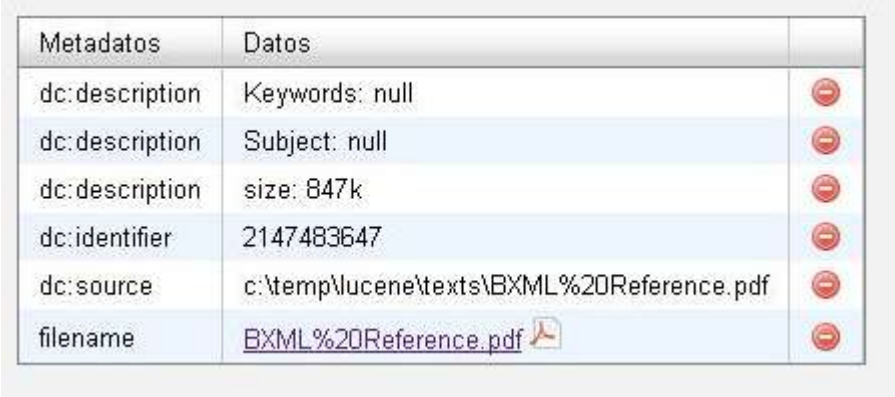


El sistema solicitará al usuario donde desea descargar el documento en la estación local, y allí será almacenado.



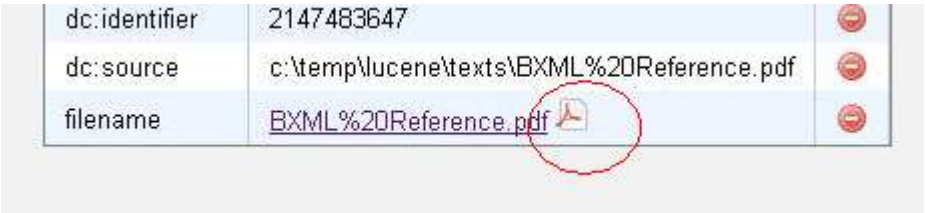
Ver detalle

En esta sección se muestran todos los datos que tiene el documento, si un dato se encuentra vacio entonces no se mostrará en el detalle.



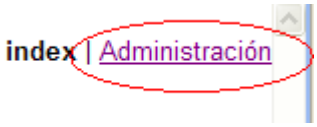
Metadatos	Datos
dc:description	Keywords: null
dc:description	Subject: null
dc:description	size: 847k
dc:identifier	2147483647
dc:source	c:\temp\lucene\texts\BXML%20Reference.pdf
filename	BXML%20Reference.pdf

Para descargar el documento se debe dar click sobre el icono del documento.



MODO ADMINISTRACIÓN

Para ingresar a este modulo se debe dar click en el link Administración ubicado en la parte superior derecha del modo de Búsqueda.



Para volver al modo de Búsqueda se debe seleccionar la opción Salir en la parte superior derecha del modo de Administración

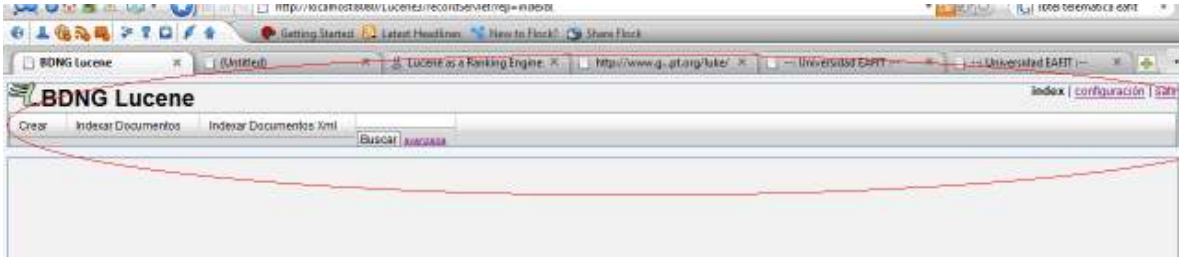
Cuando se ingresa a este modulo el sistema solicita seleccionar el repositorio en el cual se desea ejecutar la operación. También permite la opción de crear un repositorio nuevo.

El repositorio por defecto, en caso de que no se seleccione ninguno, aparecerá en la parte superior derecha. (Ver sección Administrar Repositorios para más detalles).

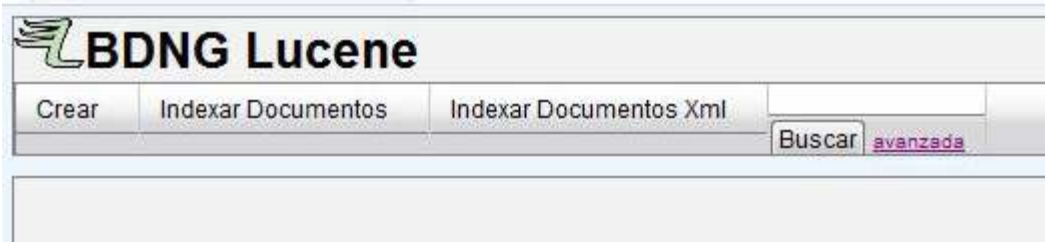
Nota: El repositorio por defecto se encuentra configurado en el archivo *configuration.properties* del sistema en la propiedad *index.* Para esto se debe tener acceso al código fuente del sistema. Para más detalles, ver el documento Manual Técnico.

MENU PRINCIPAL

El menú principal se encuentra en la parte superior de la página, parecida a una barra de herramientas.



En la parte superior izquierda se encuentran las operaciones a realizar sobre el índice seleccionado. (El índice seleccionado aparece en la parte superior derecha). Tanto las opciones para indexar como para buscar se encuentran en la misma sección.



En la parte superior derecha aparece el índice seleccionado, la opción de configuración (Administrar repositorios) y la opción de salir.

La opción de salir lleva al modo de Búsqueda.

Al seleccionar una opción el sistema direcciona al modulo correspondiente. Las instrucciones para operar sobre cada uno se encuentra en cada una de las siguientes secciones

ADMINISTRAR REPOSITORIOS

Cuando se selecciona Configurar del menú principal el sistema permite seleccionar un repositorio para indexar o buscar sobre él. También permite crear o eliminar un repositorio nuevo.

Seleccionar un repositorio

Para seleccionar un repositorio simplemente se debe hacer click sobre el nombre:

El repositorio seleccionado aparecerá en la parte superior derecha.

Crear repositorio

Para crear un repositorio nuevo se debe ingresar el nombre en el campo de texto que se encuentra debajo de todos los repositorios creados y oprimir el botón Adicionar.

Luego el sistema muestra el nuevo repositorio en la lista de repositorios creados.

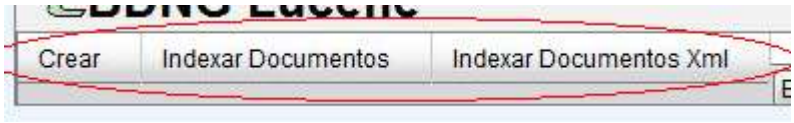
Eliminar repositorio

Para eliminar un repositorio simplemente se debe dar click en la opción eliminar como se muestra en la siguiente imagen:

El sistema solicita confirmar la eliminación del repositorio. Si el usuario confirma, se elimina el repositorio, de lo contrario vuelve a la página de Configuración.

INDEXACIÓN

Para este proceso se cuenta con el menú principal que se encuentra en la parte superior derecha con las siguientes opciones.

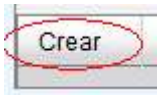


Al seleccionar una opción el sistema direcciona al modulo correspondiente. Las instrucciones para operar sobre cada uno se encuentra en cada una de las siguientes secciones

Crear un documento

Accediendo a esta opción se puede asociar un archivo con sus respectivos metadatos e indexar el contenido de dicho archivo.

Para ingresar a esta opción se debe ingresar por el menú principal.



El sistema presentará una lista con las opciones de los metadatos a ingresar y un campo en el cual el usuario deberá ingresar la información.

El usuario podrá asociar tantos metadatos como desee. Para agregar otro metadato se debe dar click en el botón “Añadir fila”.

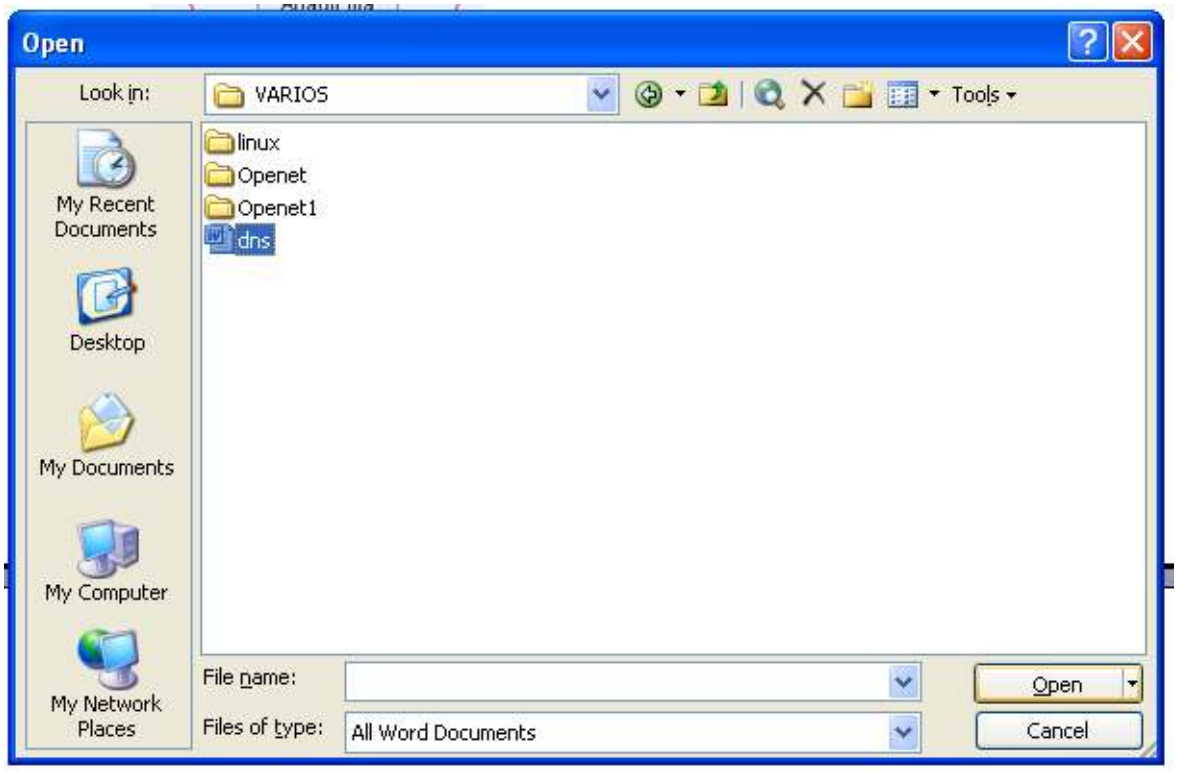
El sistema añadirá un registro para ingresar información.

Luego deberá seleccionar de la lista el metadato a asociar.

Y luego ingresar los datos correspondientes a dicho metadato y oprimir el botón “Save”.

También se cuenta con la opción de agregar un archivo digital. Para ello se debe oprimir el botón Browse.

Luego se deberá seleccionar el archivo según la ubicación en la estación de trabajo



Cuando se encuentren todos los datos listos se oprime el botón Guardar para almacenar el documento.

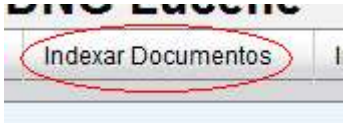
Nota: El archivo se almacenará en el servidor en la ruta configurada en el archivo *configuration.properties* en la propiedad *texto_dir*. Para esto se debe tener acceso al código fuente del sistema. Para más detalles, ver el documento Manual Técnico.

El sistema muestra si el documento fue indexado o no.

Indexar documentos

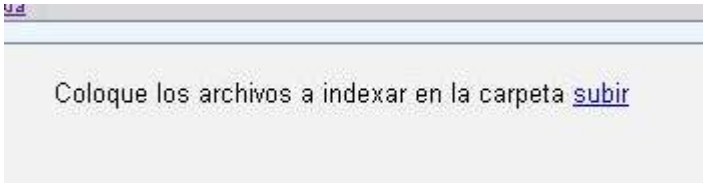
Al ingresar a esta opción se indexarán todos los documentos del directorio configurado en el archivo *configuration.properties* en la propiedad *texto_dir*.

Para ingresar a esta opción se debe ingresar por el menú principal.



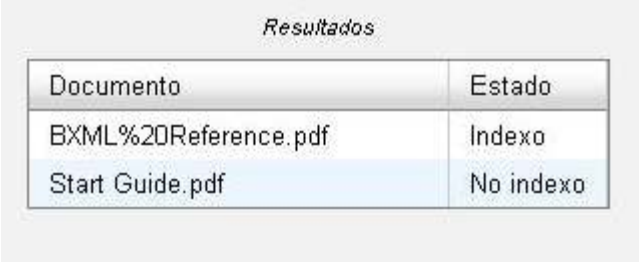
El sistema pedirá confirmar la acción. Si el usuario confirma entonces se indexarán todos los archivos contenidos en dicha carpeta y sus subcarpetas.

Se debe seleccionar el link “subir” para indexar todo el directorio.



Los archivos que indexa son los de extensión .doc, .pdf. .xls, .pps, .ppt, .html, .htm, .xhtml, .xml y .txt.

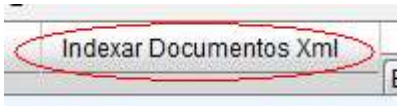
Luego de indexados todos los archivos, el sistema mostrará un listado de los documentos indexados y no indexados.



Indexar documentos XML Dublin Core

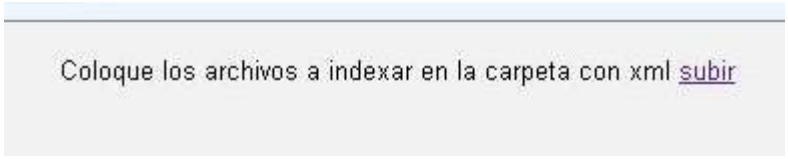
Al ingresar a esta opción se indexarán todos los documentos xml que contienen información de documentos referenciados por medio del modelo Dublin Core. El directorio a indexar será el directorio configurado en el archivo *configuration.properties* en la propiedad *texto_xml*.

Para ingresar a esta opción se debe ingresar por el menú principal.

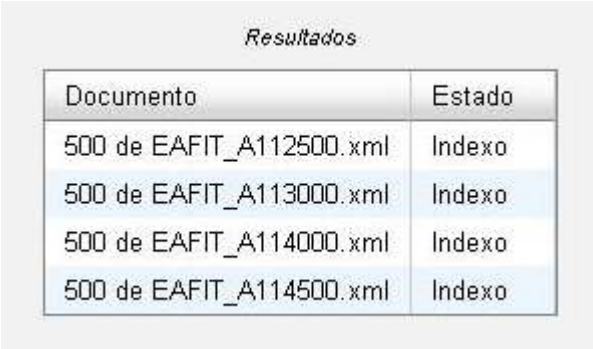


El sistema pedirá confirmar la acción. Si el usuario confirma entonces se indexarán todos los archivos contenidos en dicha carpeta y sus subcarpetas.

Se debe seleccionar el link “Subir” para indexar todo el directorio.

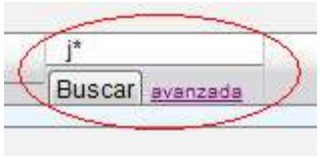


Luego de indexados todos los archivos, el sistema mostrará un listado de los documentos indexados y no indexados.



BUSQUEDA

Esta opción se encuentra en el menú principal. La búsqueda básica se puede realizar directamente desde dicho menú, o si se desea hacer una búsqueda avanzada se debe seleccionar dicha opción.

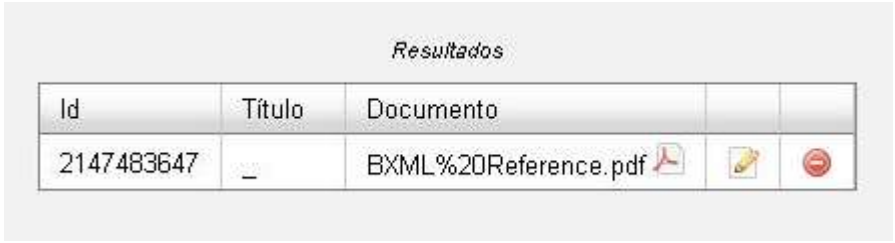


Búsqueda básica

Esta búsqueda consiste en ingresar los parámetros de búsqueda y el sistema se encargará de retornar los documentos coincidentes sin importar en que metadato o sección del documento se encuentran las palabras a buscar.

Esta búsqueda se realiza desde el menú principal, se debe ingresar la(s) palabra(s) a buscar y oprimir el botón buscar.

El sistema retornará los resultados que coincidan con esta búsqueda y los campos de Id, Titulo y Documento.

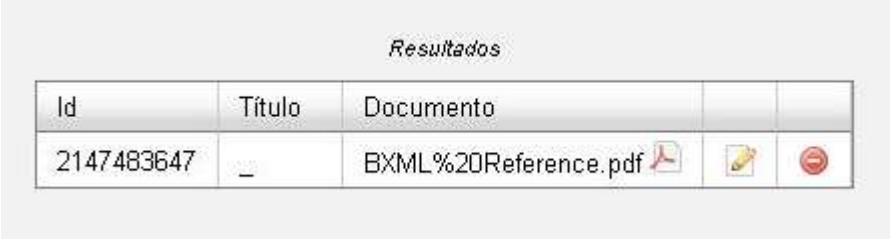


Búsqueda avanzada

Para ingresar a esta búsqueda se debe ingresar desde el menú principal.

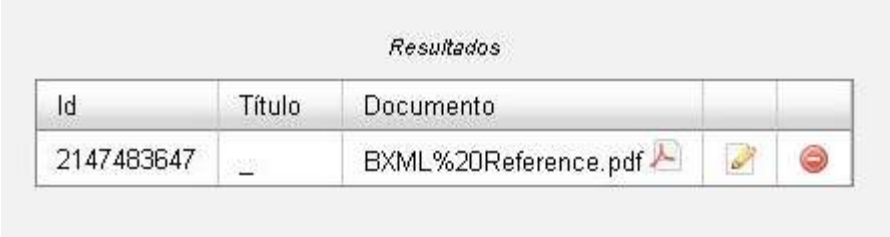
Para esta búsqueda se pueden seleccionar los campos específicos y las palabras que se desean buscar, y los operadores lógicos entre dichos campos para hacer más precisa la búsqueda.

Al oprimir el botón buscar se muestran los resultados según los criterios de búsqueda.

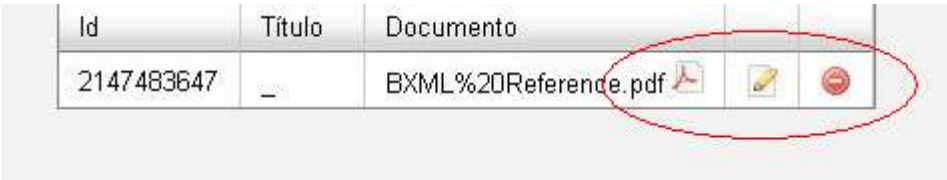


Ver resultados

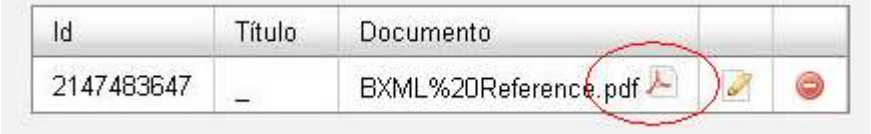
Sin importar el tipo de búsqueda, los resultados solo mostrarán algunos de los datos básicos asociados al documento.



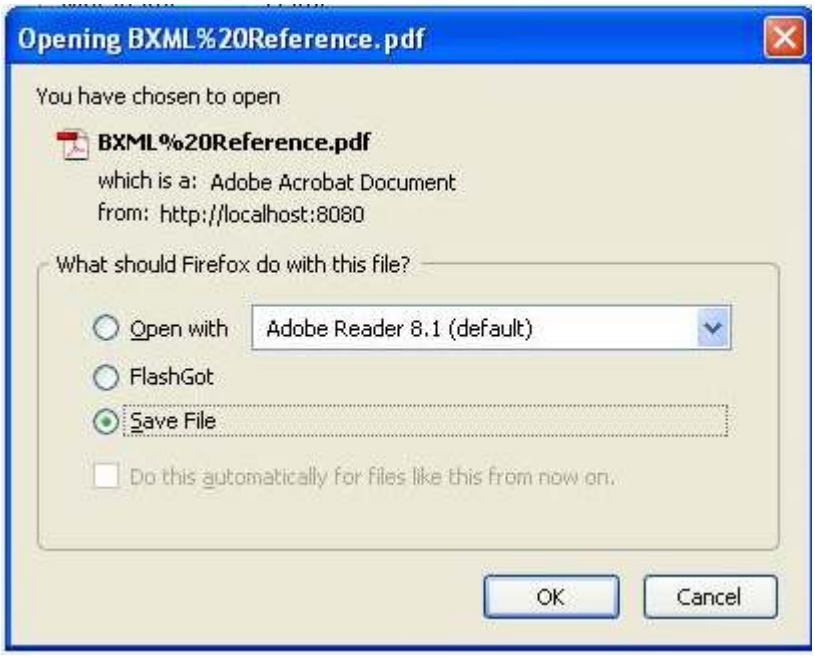
Cada registro encontrado tendrá la posibilidad de ver el detalle, descargar documento y eliminar.



Para descargar el documento se debe dar click sobre el icono del documento.

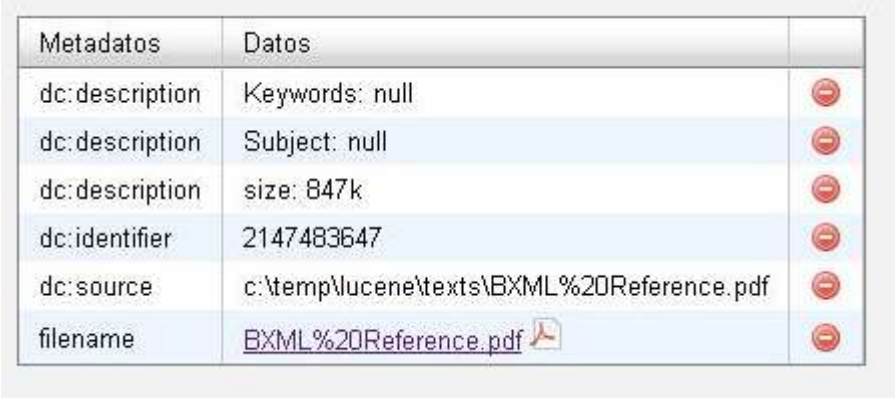


El sistema solicitará al usuario donde desea descargar el documento en la estación local, y allí será almacenado.



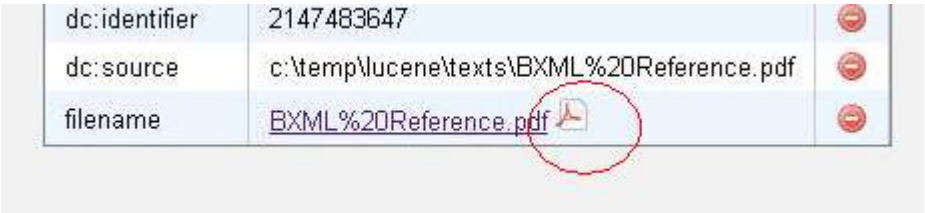
Ver detalle

En esta sección se muestran todos los datos que tiene el documento, si un dato se encuentra vacio entonces no se mostrará en el detalle.



Metadatos	Datos
dc:description	Keywords: null
dc:description	Subject: null
dc:description	size: 847k
dc:identifier	2147483647
dc:source	c:\temp\lucene\texts\BXML%20Reference.pdf
filename	BXML%20Reference.pdf

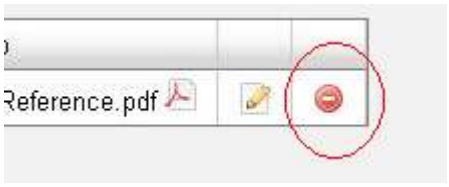
Para descargar el documento se debe dar click sobre el icono del documento.



Eliminar documento

Cuando se desea eliminar un documento se debe seleccionar la opción eliminar desde ver detalle o desde los resultados de la búsqueda:

Desde los resultados de búsqueda se debe oprimir el botón eliminar:



Desde ver detalle se debe oprimir el botón eliminar:

El sistema pide confirmación. Si el usuario confirma se elimina el documento y el archivo asociado si existe.

Manual Técnico

Sistema BDGN Lucene

CRISTIAN RÍOS VILLADA

JOHN JAIRO SILVA ZULUAGA

Medellín

Universidad Eafit

2008

TABLA DE CONTENIDO

INTRODUCCIÓN

Este documento tiene por objetivo orientar y presentar una adecuada información acerca de los componentes y relaciones entre ellos del sistema BDNG Lucene. Contiene toda la información sobre los recursos utilizados por el proyecto y una descripción detallada sobre las características físicas y técnicas de cada elemento.

Está escrito especialmente para desarrolladores interesados en conocer la estructura del BDNG Lucene con el fin de posibles mejoras o futuros desarrollos.

Nota1: No se va a detallar la estructura o código relacionado con la interfaz, pues dicho componente es independiente del núcleo del sistema.

Nota2: Se asume que el lector conoce la estructura del BDNG_Core. De lo contrario remitirse a la documentación de dicho proyecto o contactar al encargado.

REQUERIMIENTOS PARA DESARROLLO.

Plataforma de desarrollo Java. Mínimo el JDK 6 (http://java.sun.com/javase/downloads/index.jsp).

IDE: Recomendado Eclipse (http://www.eclipse.org/downloads/).

PASOS DE INSTALACIÓN PARA DESARROLLO

Esta guía de instalación es para el IDE Eclipse, ambiente en el cual fue desarrollada la aplicación.

1. Descomprimir el archivo bdng_core.zip en el workspace de Eclipse.
2. Abrir Eclipse y seleccionar el workspace en que se encuentra el bdng_core.
3. Seleccionar el proyecto bdng_core y hacer clic derecho.
4. Seleccionar Properties e ir a la sección Java Built Path.
5. Seleccionar la pestaña Libraries.
6. Oprimir el botón Add External Jars y seleccionar la carpeta del proyecto bdng_core
7. Ingresar a la carpeta Lib y agregar todas las librerías incluidas allí.
8. Oprimir el botón abrir.

Ya se encuentra el proyecto listo para analizar y modificar su código fuente.

El proyecto BDNG Lucene se encuentra en el paquete org.bdng.core.lucene

LIBRERÍAS UTILIZADAS

Lucene

Es la librería encargada del manejo de índices o repositorio y documentos. http://lucene.apache.org/

PDFBox

Librería encargada del manejo de los archivos pdf, como creación, extracción de información entre otros. http://www.pdfbox.org/

POI

Librería encargada del manejo de archivos de Office(doc, xls, pps), como la creación, edición y extracción de información. http://poi.apache.org/

FontBox

Librería que trabaja con la librería de PdfBox, y se encarga de leer los diferentes tipos de fuentes de los archivos pdf para desencriptar su texto correctamente. http://sourceforge.net/projects/fontbox/

REQUISITOS

Desde la concepción del proyecto ya se conocían las funcionalidades generales del sistema deseado, sin embargo, era necesario aclarar las necesidades que se tenían de una forma más detallada. Pero dadas las condiciones del sistema a través de sus etapas de pruebas y versiones, algunos de los requisitos fueron modificados para un producto final más óptimo.

A continuación se detallan los requisitos iniciales estructurados por funcionalidades, y finalmente se documentan los cambios hechos en dichos requisitos.

Requisitos Funcionales

Modulo Manejo de Documentos

Este modulo se encarga de describir las acciones de lo que puede hacer el usuario con un documento: Crear, actualizar y eliminar.

- Crear Documento

Descripción y Prioridad

> El sistema debe permitir ingresar la información de un documento, conformada por el contenido de texto de un archivo, la ruta del archivo, y los metadatos asociados a dicho archivo.

Para lograr esto, se presenta al usuario una pantalla en la cual se despliega la opción de diligenciar cada uno de los metadatos que el usuario desee asignar a dicho documento. La lista de metadatos, apoyados en el Dublin Core, se puede ver en Requisito de Información INF1 y analizar en más detalle las posibilidades y/o restricciones de cada uno de estos en la sección Indexación de Metadatos.

Luego de que el usuario ingrese los datos que desee, el sistema realizará los correspondientes procesos de indexación y almacenamiento (Ver Modulo Indexación).

Cada que se ingrese un nuevo documento, se optimizará automáticamente la base de datos Lucene.

Es muy importante tener en cuenta que los archivos no son requeridos en un documento.

Flujo de Eventos

- El usuario ingresa al sistema BDNG.
- El sistema despliega los campos para ingresar los metadatos según Dublin Core (Requisito de información INF1)
- El usuario ingresa los campos que desee.
- El sistema realiza las validaciones según el caso que se presente (modulo indexación)
- Si el sistema valida los datos ingresados, el documento es indexado y la base de datos optimizada.
- Si el sistema no valida los datos, se notifica la razón de por qué no se puede indexar el documento.

Requisitos Funcionales

REQ1.1.1 El sistema debe permitir el ingreso de datos y archivos al sistema para que sean indexados.

REQ1.1.2 El sistema no debe requerir agregar un archivo, ya que no es necesario que un documento tenga un archivo físico asociado.

- Actualización de Documentos.

Descripción y Prioridad

Luego de que se presenten los resultados de la búsqueda, el sistema deberá permitir al usuario actualizar el documento.

Para cada resultado se presentará la opción de Actualizar la información contenida en cada campo del documento. Si se selecciona esta opción, el sistema deberá desplegar cada uno de los campos que contienen información asociada a dicho documento. Sin restricción alguna cada campo podrá ser editado y podrán añadirse más campos o eliminarse si se desea. Para eliminar un campo solo basta con eliminar el dato actual y dejarlo en blanco.

Luego de editada la información, el sistema reindexará nuevamente el documento y se optimizará la base de datos Lucene.

Flujo de Eventos

- El sistema despliega los resultados de la búsqueda, con la opción de editar cada documento.
- El usuario selecciona la opción de editar un determinado documento.

- El sistema despliega la información de dicho documento, siendo cada uno de los campos editables.
- Si el usuario deja un campo en blanco, dicho campo se eliminará del documento.
- Si el usuario lo desea, puede agregar un nuevo campo y añadirle información.
- El usuario actualiza los datos y selecciona la opción Guardar.
- Si el sistema valida los datos ingresados, el documento es indexado y la base de datos optimizada.
- Si el sistema no valida los datos, se notifica la razón de por qué no se puede indexar el documento.

Requisitos Funcionales

REQ1.1.3 El sistema debe permitir la actualización de documentos ya existentes, permitiendo también la eliminación o agregación de nuevos campos a dicho documento.

- Eliminación de Documentos.

Descripción y Prioridad

Luego de que se presenten los resultados de la búsqueda, el sistema deberá permitir al usuario eliminar el documento.

Para cada resultado se presentará la opción de eliminar el documento. Si se selecciona esta opción, el sistema deberá preguntar al usuario si está seguro de eliminar el documento. Si el usuario cancela, vuelve a los

resultados de la búsqueda, si el usuario confirma, dicho documento será eliminado.

Luego de eliminado el documento, se optimizará la base de datos Lucene.

Flujo de Eventos

- El sistema despliega los resultados de la búsqueda, con la opción de eliminar cada documento.
- El usuario selecciona la opción de eliminar un determinado documento.
- El sistema pregunta al usuario si está seguro de eliminar el documento.
- Si el usuario cancela la acción, el sistema vuelve a mostrar los resultados de la búsqueda.
- Si el usuario confirma la eliminación, el documento es eliminado y la base de datos optimizada.
- Si el sistema no permite la eliminación, se notifica la razón de por qué no se puede eliminar el documento.

Requisitos Funcionales

REQ1.1.4 El sistema debe permitir la eliminación de documentos ya existentes.

Modulo Indexación

Este módulo se encarga de describir los procesos necesarios para la indexación de documentos y archivos.

- Almacenamiento de Archivos en un repositorio

Descripción

Cada que se crea o edita un documento, y éste tiene un archivo asociado, el sistema deberá almacenar dicho archivo en un repositorio definido en el servidor, para luego extraer el texto según el tipo de archivo, ser indexado y asociar el índice al documento.

Flujo de Eventos

- El usuario crea un documento con un archivo asociado.
- El sistema carga dicho archivo y lo almacena en el repositorio del servidor.
- El sistema extrae el texto del archivo.
- El sistema indexa el contenido del texto y lo asocia a dicho documento.

Requisitos Funcionales

REQ1.2.1 El Sistema deberá permitir el almacenamiento de archivos en el repositorio del servidor para luego ser indexados.

- Indexación de Archivos

Descripción

El sistema deberá permitir la indexación del contenido de texto de los archivos asociados a un documento. Este proceso se podrá realizar de 2 formas:

- Indexación de un único archivo: Si al crear un documento se hace referencia a un archivo específico, se almacena dicho archivo en la carpeta del servidor y se procede a su indexación.
- Indexación de múltiples archivos: Se permitirá indexar archivos en batch, es decir, múltiples archivos al mismo tiempo seleccionando la ruta del directorio. También indexará los documentos en todos los subdirectorios.
- Indexación de múltiples archivos Dublin Core: Se debe tener la capacidad de indexar los documentos xml con el esquema Dublin Core.

Flujo de Eventos

- El usuario ingresa al modulo crear documento.
- El usuario ingresa los metadatos asociados al documento
- El usuario selecciona uno o mas archivos asociados al documento.
- El sistema carga dichos archivos y los almacena en el repositorio del servidor.
- El sistema extrae el texto de los archivos.
- El sistema indexa el contenido del texto y lo asocia a dichos metadatos.

Requisitos Funcionales

REQ1.2.2 El sistema deberá permitir la indexación de uno o más archivos asociados a los mismos metadatos.

- Indexación de diferentes tipos de archivo

Descripción y Prioridad

El sistema deberá indexar diferentes tipos de archivos que manejen texto en su contenido. Primero se deberá extraer el texto en su interior y luego indexarse.

Los archivos más comunes que contienen texto y deben ser indexados son los que poseen las siguientes extensiones:

- .PDF (Acrobat Reader)
- .DOC (Microsoft Word)
- .PPT o .PPS (Microsoft PowerPoint)
- .XML
- .HTML

Flujo de Eventos

- El usuario crea un documento y asocia un archivo.
- El sistema almacena el archivo en el repositorio.
- El sistema identifica el tipo de archivo.
- El sistema extrae el texto.
- El sistema indexa el texto y lo asocia al documento.

Requisitos Funcionales

REQ1.2.3 El sistema debe indexar los diferentes tipos de archivos más comunes que manejen texto en su contenido.

- Indexación de Metadatos

Descripción y Prioridad

El sistema deberá permitir asociar a cada documento los campos contenidos en el requisito de información INF1, los cuales pertenecen al modelo Dublin Core.

Solo el campo DC.Identifier es requerido, los demás son opcionales, incluyendo el archivo. Un documento puede tener la información de los metadatos completa pero sin existir un archivo físico asociado.

Un documento también puede tener el mismo metadato en varias ocasiones. Por ejemplo un documento puede tener más de un autor, o más de una fuente.

Flujo de Eventos

-

Requisitos Funcionales

REQ1.2.4 El sistema debe permitir asociar a un documento, los metadatos según el modelo Dublin Core.

REQ1.2.5 Un documento puede tener asociado el mismo campo más de una vez.

Modulo de búsquedas

Este módulo describe los tipos de búsqueda que debe soportar la aplicación.

- Búsqueda sencilla

Descripción y Prioridad

El sistema debe permitir realizar búsquedas de los datos indexados ingresando la(s) palabra(s) a buscar sin importar en que metadato se encuentre. Es decir, el usuario solo se encargará de ingresar la(s) palabra(s) a buscar y el sistema se encargará de buscar dicho parámetro en todos los campos de todos los documentos.

Si se desea buscar un documento en un campo específico debe ir a Búsqueda Avanzada.

Flujo de Eventos

- El usuario ingresa al módulo de búsquedas.
- El usuario ingresa las palabras de búsqueda contenidas en un solo campo.
- El sistema despliega los resultados de búsqueda.

Requisitos Funcionales

REQ1.3.1 El sistema debe permitir búsquedas sencillas.

- Búsqueda Avanzada

Descripción y Prioridad

El sistema deberá permitir búsquedas por múltiples campos con el fin de que el sistema retorne resultados más exactos de lo q necesite el usuario.

Por ejemplo, si se desea buscar los libros de un autor específico que contengan determinadas palabras o temas, el sistema se encargará de realizar los filtros necesarios para retornar resultados más precisos.

Flujo de Eventos

- El usuario ingresa al módulo de búsquedas.
- El usuario ingresa las palabras de búsqueda contenidas en uno o más campos específicos.
- El sistema despliega los resultados de búsqueda.

Requisitos Funcionales

REQ1.3.2 El sistema debe permitir búsquedas por múltiples campos.

Modulo de resultados y Visualización de Documentos.

Este módulo comprende las opciones y funcionalidades que debe tener el sistema luego de que se encuentren los datos y deban ser mostrados al usuario.

- Visualización de Resultados

Descripción y Prioridad

Esta sección se encargará de mostrar los resultados de las búsquedas. Por defecto se mostrarán primero los archivos con más relevancia o mayor número de ocurrencias.

Los resultados mostrarán uno a uno los documentos con sus respectivos campos, es decir, se mostrará solo la información de los campos que se

encuentren diligenciados. No se mostrarán los campos que fueron guardados en blanco.

Adicionalmente, cada documento tendrá la opción de actualizar y eliminar.

Si existe un archivo asociado al documento, el usuario podrá descargarlo o abrirlo en el formato original.

Flujo de Eventos

- El usuario realiza la búsqueda.
- El sistema despliega los resultados, únicamente con la información de los campos que han sido ingresados y las opciones de Actualizar y Eliminar.
- Si el usuario selecciona Actualizar ir a Actualización de Documentos.
- Si el usuario selecciona Eliminar ir a Eliminación de Documentos.
- Si el documento tiene un archivo asociado, el sistema muestra la opción de descargarlo o verlo en formato web.
- Si el usuario selecciona descargar el archivo, ir a - Descarga de archivos según el tipo de archivo.

Requisitos Funcionales

REQ1.4.1 El sistema debe mostrar los resultados de búsqueda, mostrando en orden los documentos de mayor a menor relevancia con las opciones de actualizar y eliminar documentos.

- Descarga de archivos según el tipo de archivo

Descripción y Prioridad

Esta sección se encarga de permitirle al usuario descargar o abrir el archivo asociado a un documento en el momento de que el sistema muestra los resultados de la búsqueda.

Flujo de Eventos

- El sistema despliega los resultados de la búsqueda.
- El usuario selecciona la opción de descargar archivo.
- El sistema pregunta al usuario si desea abrir o descargar el documento.
- El usuario elige la opción.
- El sistema descarga el archivo en el repositorio del usuario.

Requisitos Funcionales

REQ1.4.2 El sistema debe permitir descargar o abrir el archivo asociado a un documento.

Administración de Repositorios

Descripción y Prioridad

El sistema debe permitir la creación y eliminación de repositorios Lucene, con el fin de crear varios repositorios que permitan categorizar los documentos. Es decir, se puede crear un repositorio para documentos de un tipo o un usuario especifico, y permitir que un usuario seleccione el repositorio sobre el cual desea indexar o buscar.

Flujo de Eventos

- El usuario ingresa al Administrador de repositorios.
- El usuario selecciona la opción de crear repositorio.
- El usuario ingresa el nombre del repositorio y oprime Guardar.
- El sistema crea los archivos necesarios para el repositorio.
- El usuario selecciona la opción de eliminar repositorio.
- El sistema elimina el repositorio.
- El usuario selecciona un repositorio específico.
- El sistema configura el repositorio para buscar e indexar sobre el.

Requisitos Funcionales

REQ1.5.1 El sistema debe permitir la creación, eliminación y selección de repositorios.

Requisitos No-Funcionales

Los Atributos de calidad son aspectos no funcionales pero de vital importancia para descripción de la arquitectura de un sistema estos son: Confiabilidad, seguridad, usabilidad, desempeño, mantenibilidad.

Requisitos de Información

INF1 Define los campos definidos por el Dublin Core. Ver tabla 2.

Tabla 1. Requisito de Información 1 (INF1). Estructura Dublin Core.

Campo	Req.	Tipo de Campo	Validación
DC.Title		Texto	**Título:** el nombre dado a un recurso, habitualmente por el autor.
DC.Subject		Texto	**Claves:** los tópicos del recurso. Típicamente, Subject expresará las claves o frases que describen el título o el contenido del recurso. Se fomentará el uso de vocabularios controlados y de sistemas de clasificación formales.
DC.Description		Texto	**Descripción:** una descripción textual del recurso. Puede ser un resumen en el caso de un documento o una descripción del contenido en el caso de un documento visual.
DC.Source		Texto	**Fuente:** secuencia de carácteres usados para identificar unívocamente un trabajo a partir del cual proviene el recurso actual.
DC.Language		Texto	**Lengua:** lengua/s del contenido intelectual del recurso.
DC.Relation		Texto	**Relación:** es un identificador de un segundo recurso y su relación con el recurso actual. Este elemento permite enlazar los recursos relacionados y las descripciones de los recursos.
DC.Coverage		Texto	**Cobertura:** es la característica de cobertura espacial y/o temporal del contenido intelectual del recurso. La cobertura espacial se refiere a una región física, utilizando por ejemplo coordenadas. La cobertura temporal se refiere al contenido del recurso, no a cuándo fue

			creado (que ya lo encontramos en el elemento Date).
DC.Creator		Texto	**Autor o Creador:** la persona o organización responsable de la creación del contenido intelectual del recurso. Por ejemplo, los autores en el caso de documentos escritos; artistas, fotógrafos e ilustradores en el caso de recursos visuales.
DC.Publisher		Texto	**Editor:** la entidad responsable de hacer que el recurso se encuentre disponible en la red en su formato actual.
DC.Contributor		Texto	**Otros Colaboradores:** una persona u organización que haya tenido una contribución intelectual significativa, pero que esta sea secundaria en comparación con las de las personas u organizaciones especificadas en el elemento Creator. (por ejemplo: editor, ilustrador y traductor).
DC.Rights		Texto	**Derechos:** son una referencia (por ejemplo, una URL) para una nota sobre derechos de autor, para un servicio de gestión de derechos o para un servicio que dará información sobre términos y condiciones de acceso a un recurso.
DC.Date		Texto	**Fecha:** una fecha en la cual el recurso se puso a disposición del usuario en su forma actual. Esta fecha no se tiene que confundir con la que pertenece al elemento Coverage, que estaría asociada con el recurso en la medida que el contenido intelectual está de alguna manera relacionado con aquella fecha.
DC.Type		Texto	**Tipo del Recurso:** la categoría del recurso. Por ejemplo, página personal, romance, poema,

			diccionario, etc.
DC.Format		Texto	**Formato:** es el formato de datos de un recurso, usado para identificar el software y, posiblemente, el hardware que se necesitaría para mostrar el recurso.
DC.Identifier	X	Texto	**Identificador del Recurso:** secuencia de carácteres utilizados para identificar unívocamente un recurso. Ejemplos para recursos en línea pueden ser URLs i URNs. Para otros recursos pueden ser usados otros formatos de identificadores, como por ejemplo ISBN ("International Standard Book Number").

Requisitos de Confiabilidad

CON1 Borrado físico – lógico: El aplicativo permitirá el borrado físico de elementos siempre que no se viole integridad referencial, de lo contrario, pondrá el ítem en un estado en el que se reconozca como inactivo para que no se use en la operación normal y sólo sea accesible para las consultas y reportes.

Requisitos de Desempeño

DES1 El sistema debe manejar una concurrencia de al menos 50 usuarios.

DES2 Cada que haya una creación, actualización o eliminación de documentos se debe optimizar la base de datos.

Requisitos de Interfaces de Software

ISW1 El sistema se debe integrar con la librería BDNG_CORE y sus métodos abstractos.

ISW2 Browser: La aplicación deberá correr en Internet Explorer 6.0 o superior y con Resolución gráfica de 1024 x 768 y en Mozilla Firefox 1.5 o superior y con Resolución gráfica de 1024 x 768.

Requisitos de Usabilidad

USA1 Al crear o editar se deben poder agregar más campos al documento.

DISEÑO

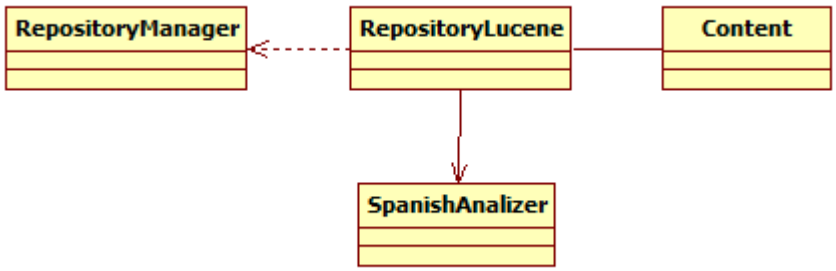
A alto nivel, la BDNG Lucene es una aplicación sencilla y fácil de entender, por eso esta sección se encargará de detallar cada uno de los 2 procesos básicos del sistema: Indexación y Búsqueda. Adicionalmente se explicará la funcionalidad de configuración de los repositorios como un módulo extra, pero que dada su sencillez no se entrará mucho en detalle.

Cada módulo contendrá el diagrama de casos de uso a alto nivel y un diagrama de actividades de cada una de las operaciones básicas del sistema.

Finalmente se encuentra una explicación detallada y las líneas de código (métodos y parámetros) utilizadas en cada una de estas actividades.

Pero antes de comenzar, se explicará brevemente la estructura general del sistema Bdng Lucene.

DIAGRAMA DE CLASES GENERAL



RepositoryManager

Se encargar de administrar el acceso a los índices creados por Lucene. Contiene los siguientes métodos:

- **createRepository** (String *index*): Crea el repositorio *index*.
- **deleteRepository** (String *Index*): Elimina el repositorio *index*.
- **listRepository** (): Método que lista los repositorios Lucene ya creados.
- **startIndexer**(): Inicializa el repositorio por defecto en modo escritura
- **startIndexer**(String *index*): Inicializa el repositorio *index* en modo escritura
- **startReader**():Inicializa el repositorio por defecto en modo lectura
- **startReader**(String *index*): Inicializa el repositorio *index* en modo lectura
- **startSearcher**(): Inicializa el repositorio por defecto en modo búsqueda
- **startSearcher**(String *index*): Inicializa el repositorio *index* en modo búsqueda

RepositoryLucene

Se encarga de realizar las operaciones sobre el índice, buscar, indexar, actualizar y eliminar. Es la clase que implementa los metodos del Bdng Core. Contiene los siguientes métodos:

- **delete**(String *idrec*): Elimina un registro del repositorio según el *idrec.* Retorna falso si el documento no fue indexado.
- **getInstance**():Inicializa el repositorio
- **numRecords**():Entrega el número de registros en el repositorio
- **query**(int *idcampo*, String *value*): Devuelve una búsqueda hecha al repositorio, según el nombre del campo en el cual se desea buscar y el valor a buscar. Retorna un array de IRecord.

- **query**(String *strQuery*): Devuelve una busqueda hecha al repositorio en el lenguaje propio de Lucene. Retorna un array de IRecord.
- **retrieve**(String *idrec*): Retorna un registro IRecord dado por su identificador Lucene.
- **store**(File f): Almacena los registros que hay en un documento xml dublin core, en el repositorio. Primero extrae el contenido de cada documento, luego los almacena en un array y finalmente se indexa. Retorna "-1" si no se indexó el documento.
- **store**(IRecord *rec*): Almacena un IRecord en el repositorio.

  Para identificar cada documento almacenado, al momento de indexar se le crea un identificador único independiente del dc:Identifier. Este identificador se utilizara solo por el BDNG Lucene y no tiene relevancia para las demás interfaces del BDNG ni para el usuario, se utiliza principalmente para identificar cada documento Lucene. La relación entre dicho documento y los metadatos en otra interfaz (eXist o MySQL) se hará directamente por el dc:identifier.

  Puede indexar de 3 modos: Solo metadatos, solo contenido textual de un archivo, y metadatos y contenido textual). Para leer el tipo de modo a indexar se verifica la propiedad configurada en el archivo configuración.properties.

  Si se selecciona el segundo modo, se podrán ingresar metadatos pero serán almacenados en una archivo xml y se podrán relacionar dichos metadatos con el documento Lucene (contiene el indice del contenido textual del archivo) por medio del id único.

  Retorna "True" si no se indexó el documento.
- **update**(IRecord *reg*): Actualiza un registro IRecord del repositorio. Retorna falso si no se actualizó el documento.

- **update**(String *idReg*, IRecord *reg*): Actualiza un registro IRecord del repositorio por el identificador. Retorna falso si no se actualizó el documento.

Content

Se encarga de identificar el tipo de archivo y extraer el texto para enviarlo a RepositoryLucene quien se encarga de indexarlo. Sus métodos son los siguientes:

- **ObtenerContenido** (Document *document*, String *filename*, String id): Se encarga de leer la extensión del archivo y dependiendo de ésta se delega la extracción de texto al método correspondiente. Retorna falso si no se pudo extraer el contenido.
- **AddContentXmlDC** ():Extrae el contenido de los archivos xml dublic core
- **AddContentTxt** ():Extrae el contenido de los archivos del block de notas .txt
- **AddContentDoc** ():Extrae el contenido de los archivos de Microsoft Word .doc
- **AddContentXml**():Extrae el contenido de los archivos xml
- **AddContentXls** ():Extrae el contenido de los archivos de Microsoft Excel .xls
- **AddContentPpt**():Extrae el contenido de los archivos de Microsoft PowerPoint .ppt y .pps
- **AddContentRtf** ():Extrae el contenido de los archivos de Microsoft Word .rff
- **AddContentHtm**():Extrae el contenido de los archivos web .htm, .htlml y .xhtml.
- **AddContentPdf**():Extrae el contenido de los archivos .pdf

SpanishAnalizer

Es una interfaz de Lucene que se encarga de separar las Stop Words o palabras que no se deben indexar. Lo característico es que son palabras en español y en inglés configuradas en el archivo configuration.properties. Se pueden agregar mas palabras si el sistema lo requiere.

INDEXACIÓN

Indexar es almacenar un elemento y que permanezca identificado de tal manera que el componente que vaya a buscarlo lo encuentre directamente o no tenga que indagar mucho ni gastar mucho tiempo para localizarlo.

El proceso de indexación va desde el momento en que el usuario carga un documento y/o ingresa los datos hasta que el sistema almacena el índice en un repositorio.

En este proyecto, Lucene se encarga directamente de crear dicho índice y almacenarlo en un repositorio que el usuario determina. Además, indexa el contenido de texto completo de archivos y no solo sus metadatos asociados.

El sistema permite también la opción de Indexar un directorio completo. Es decir, almacena e indexa cada uno de los archivos digitales que contienen texto y pueden ser leídos por el sistema, almacenando en un repositorio determinado tanto el índice como los archivos indexados.

El sistema esta desarrollado suponiendo que pueden existir identificadores Dublin Core (dc:identifier) repetidos, por ejemplo al momento de ingresar una tesis que contiene varios documentos, el usuario ingresará el mismo identificador porque todos componen un mismo elemento, sin embargo cada documento se generará un identificador único, independiente del Dublin Core Identifier. En el caso de que se presente un nuevo documento a indexar con un identificador que ya exista, el sistema generará una excepción que indicará al usuario que dicho documento no se creará porque ya existe y si desea agregarlo debe realizar una actualización.

Diagramas de decision y explicación si hay ids repetidos.

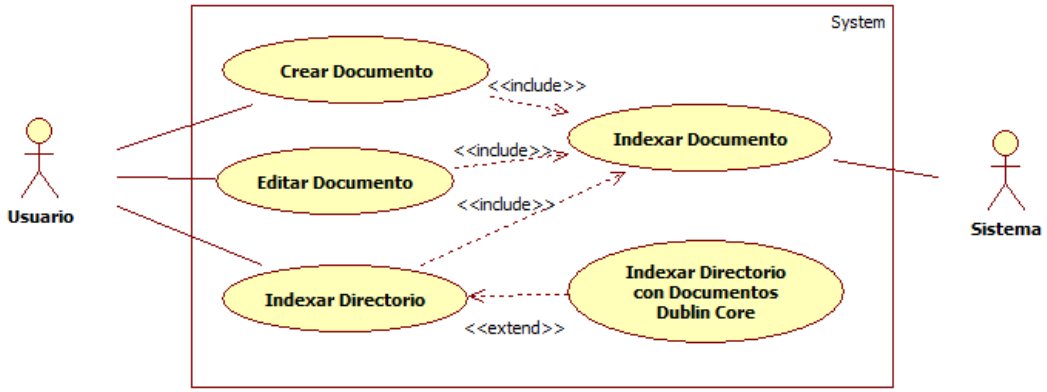
Dado que la fortaleza de Lucene es indexar el contenido textual de los archivos, BDNG Lucene tiene la opción de indexar documentos de tres modos diferentes para probar su funcionalidad en BDNG (Este modo se configurará en la propiedad modo del archivo configuración.properties):

- Indexar Metadatos y el contenido de su archivo asociado: En este caso, los metadatos y el contenido textual se indexan en un mismo documento Lucene.

- Indexar solo Metadatos: En este caso se creará un documento Lucene solo con metadatos. Similar a lo que hacer eXist. Se puede hacer referencia a un archivo, pero éste no será indexado.

- Indexar solo el contenido de su archivo asociado: En este caso se creará un documento Lucene solo con el contenido del archivo cargado. Pero este documento deberá tener un identificador que lo relacione con el registro en el repositorio de metadatos.

Diagrama de Casos de uso



Crear Documento-Indexar Documento

Descripción:

Se incluye en un mismo caso de uso ambos procesos Crear Documento e Indexar documento, pues ambos se ejecutan cuando el usuario decide indexar un archivo digital.

El usuario puede cargar un archivo y asociarle los metadatos definidos por Dublin Core. Un documento puede tener a su vez tantos metadatos como el usuario desee. Es decir, no hay problema si el archivo contiene2 autores y el usuario desea ingresarlos como metadatos independientes.

Se indexan tanto los metadatos como el contenido del archivo digital.

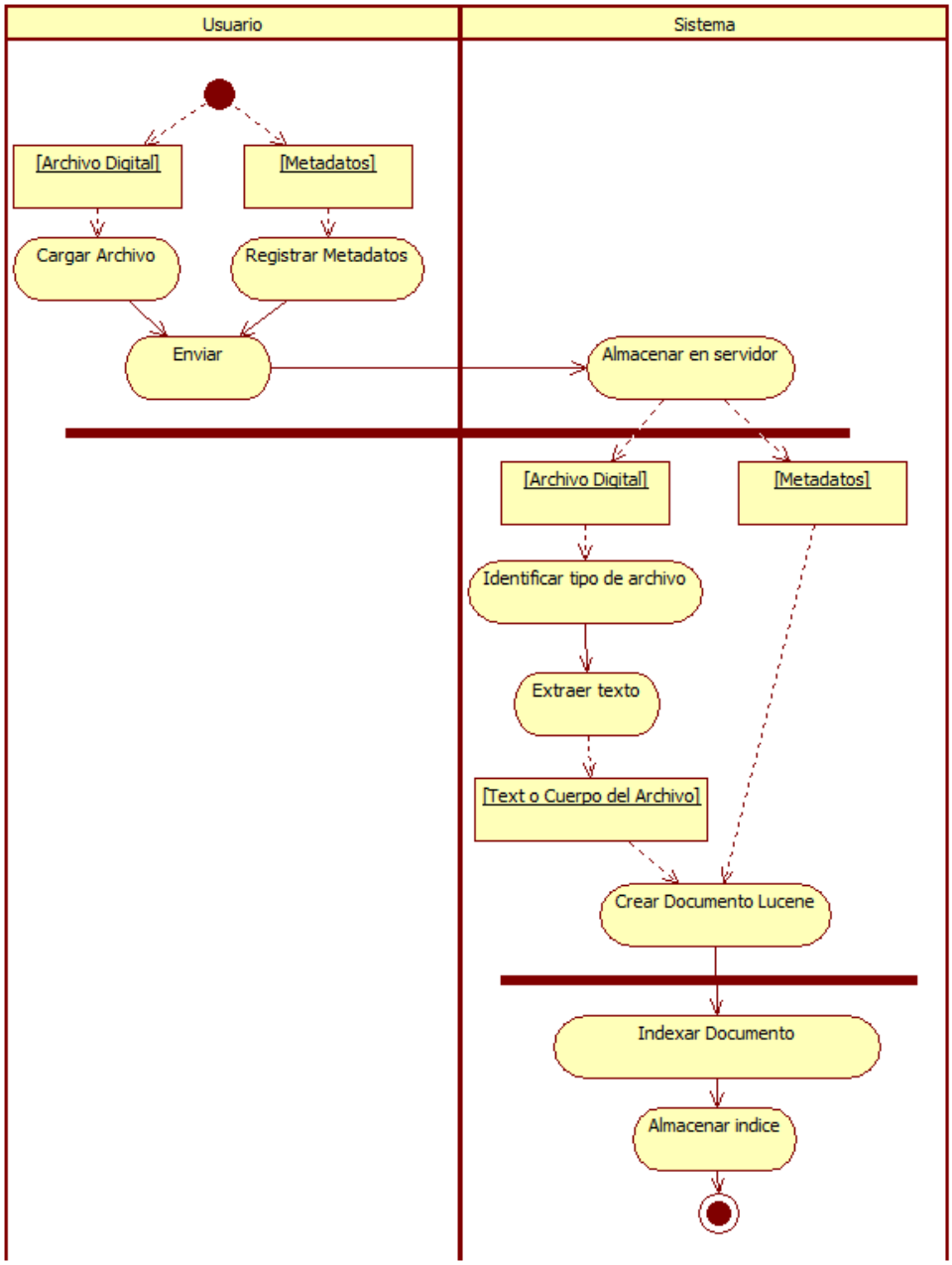
Precondiciones:

- Ya debe existir al menos un repositorio Lucene (Indice) en el sistema y configurado en el configuration.properties.

Postcondiciones:

- Se indexan tanto los metadatos como el contenido del archivo
- Se almacena el archivo digital en el repositorio de archivos

Flujo de Eventos



Especificación

Actividad	**Descripción**
Cargar Archivo	El usuario se encarga de seleccionar un archivo y cargarlo al servidor. Esto hace parte de la interfaz.
Registrar Metadatos	El usuario se encarga de seleccionar los metadatos e ingresarle los datos asociados a un archivo. Esto hace parte de la interfaz
Enviar	El usuario confirma la indexación.
Almacenar en el servidor	El sistema recibe el archivo y los metadatos y los almacena en el repositorio. Se almacenan en una carpeta temporal texto_dir configurada en configuration.properties.
Identificar tipo de archivo	Se lee la extensión del archivo. La clase Content con el método obtenerContenido. Y dependiendo de la extensión identificada se dirige al método especifico en la misma clase para extraer el texto
Extraer Texto	La clase Content posee un método diferente para extraer el contenido textual de los archivos digitales según su extensión
Crear Documento Lucene	El texto extraído se almacena en el campo content del documento lucene, y se asigna cada metadato al campo correspondiente. Este proceso se encuentra en el método store en la clase RepositoryLucene.
Indexar Documento	Para indexar el documento se utiliza la clase

	RepositoryLucene y el método Store.
Almacenar índice	Se almacena el documento en el índice. La clase RepositoryLucene el método index. El cual es un método sincronizado para evitar problemas de concurrencia.

Indexar Directorio

Descripción:

Este caso de uso se encarga de indexar todos los archivos de las extensiones permitidas (pdf, xls, doc, txt, xml) contenidos en un directorio.

El sistema lee archivo por archivo, indexa su contenido y si el proceso es exitoso almacena también el archivo en un repositorio de archivos. Si el archivo no pudo ser leído o indexado informará al usuario.

Adicionalmente, indexara los subdirectorios contenidos en dicho directorio realizando el mismo proceso.

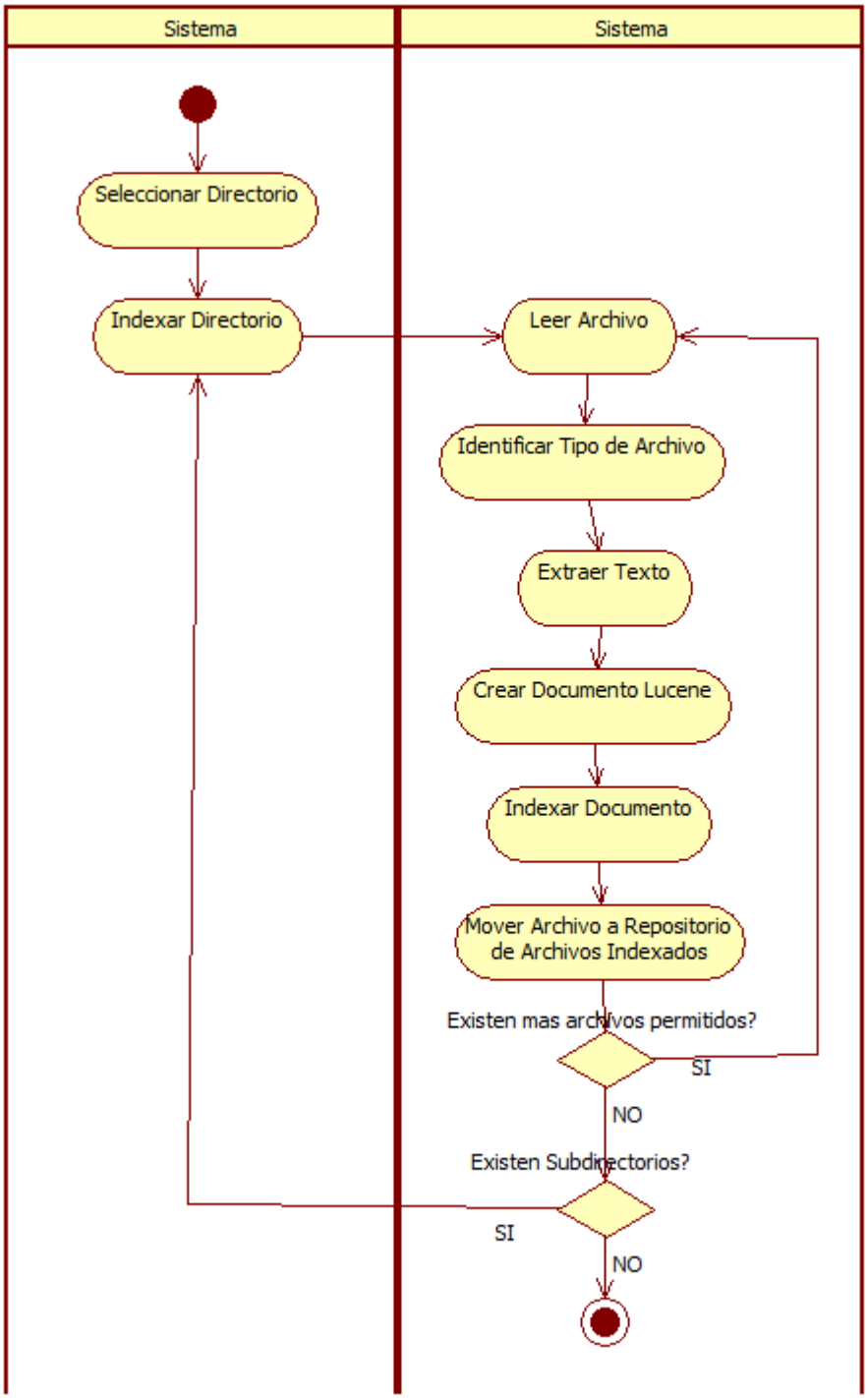
Precondiciones:

- Ya debe existir al menos un repositorio Lucene (Indice) en el sistema y configurado en el configuration.properties.
- El Directorio a indexar debe estar configurado en el configuration.properties y debe ser un directorio del servidor, no puede ser un directorio remoto.

Postcondiciones:

- Se indexa el contenido de los archivos que se encuentran dentro del directorio.
- Se indexan los archivos contenidos en los subdirectorios.
- Se almacena cada archivo digital indexado en el repositorio de archivos

Flujo de Eventos



Especificación

Actividad	**Descripción**
Seleccionar directorio	El usuario se encarga de seleccionar un directorio para indexar. Se configura en el configuration.properties la propiedad texto_dir.
Indexar Directorio	El usuario confirma que desea indexar el directorio.
Leer Archivo	El sistema identifica un archivo.
Identificar tipo de archivo	Se lee la extensión del archivo. La clase Content con el método obtenerContenido. Y dependiendo de la extensión identificada se dirige al método especifico en la misma clase para extraer el texto
Extraer Texto	La clase Content posee un método diferente para extraer el contenido textual de los archivos digitales según su extensión
Crear Documento Lucene	El texto extraído se almacena en el campo content del documento lucene, y se asigna cada metadato al campo correspondiente. Este proceso se encuentra en el método store en la clase RepositoryLucene.
Indexar Documento	Para indexar el documento se utiliza la clase RepositoryLucene y el método Store.
Almacenar índice	Se almacena el documento en el índice. La clase RepositoryLucene el método index. El cual es un método sincronizado para evitar problemas de concurrencia.
Mover Archivo a repositorio de archivos	Se mueve el archivo indexado a un repositorio de archivo configurado y se elimina de la carpeta de

indexados	origen. Esta acción se realiza desde el método store de la clase RepositoryLucene, solo si el documento fue indexado correctamente.

Indexar Directorio con Documentos Dublin Core

Descripción:

Este caso de uso se encarga de indexar todos los archivos XML de contenido específico incluidos en un directorio. Este tipo de archivos XML contienen varios documentos etiquetados según la estructura Dublin Core, por eso deben tener un trato específico para guardar cada metadato en el lugar que le corresponda.

El sistema lee archivo por archivo, indexa su contenido y si el proceso es exitoso almacena también el archivo en un repositorio de archivos. Si el archivo no pudo ser leído o indexado informará al usuario.

Adicionalmente, indexara los subdirectorios contenidos en dicho directorio realizando el mismo proceso.

Precondiciones:

- Ya debe existir al menos un repositorio Lucene (Indice) en el sistema y configurado en el configuration.properties.
- El Directorio a indexar debe estar configurado en el configuration.properties y debe ser un directorio del servidor, no puede ser un directorio remoto.
- El archivo XML debe contener la estructura de etiquetado Dublin Core para cada documento.

Postcondiciones:

- Se indexa el contenido de los archivos que se encuentran dentro del directorio.
- Se indexan los archivos contenidos en los subdirectorios.
- Se almacena cada archivo digital indexado en el repositorio de archivos

Flujo de Eventos

Igual al proceso de Indexar directorio pero la lectura de archivos y extracción de texto se realiza de la siguiente manera:

Se encuentra en la clase Content en el método addContentXmlDC. El cual lee la estructura y si encuentra el tag rdf:description en el archivo, significa que el contenido dentro ese tag es un documento dublin core, y cada tag interno es un metadato.

Editar Documento-Indexar Documento

Descripción:

Se incluye en un mismo caso de uso ambos procesos Editar Documento e Indexar documento, pues ambos se ejecutan cuando el usuario decide editar un documento que ya se encuentra almacenado.

Este proceso se puede ejecutar luego de que el usuario ejecuta una búsqueda y decide editar el contenido de un documento.

En Lucene, no se puede actualizar directamente un documento indexado, primero se requiere recuperar la información almacenada de dicho documento, cargarla temporáneamente, borrar el documento del índice, y volver a almacenar e indexar la información cargada temporalmente más la editada. Es decir, se debe eliminar

el documento del índice y volverlo a indexar. El proceso de indexación es el mismo que al agregar documento.

El usuario puede cargar un archivo y asociarle los metadatos definidos por Dublin Core. Un documento puede tener a su vez tantos metadatos como el usuario desee. Es decir, no hay problema si el archivo contiene2 autores y el usuario desea ingresarlos como metadatos independientes.

Se indexan tanto los metadatos como el contenido del archivo digital.

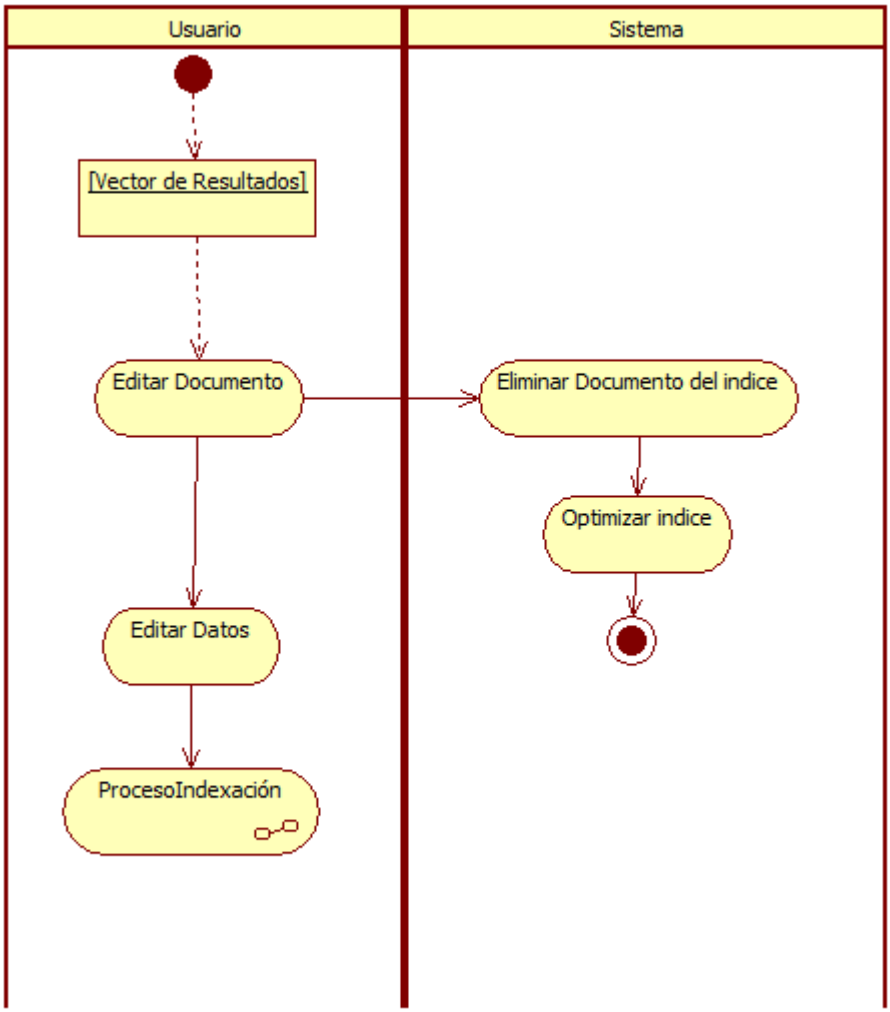
Precondiciones:

- Seleccionar un documento a editar luego del resultado de una búsqueda.
- El documento existe en el índice

Postcondiciones:

- Se indexan tanto los metadatos como el contenido del archivo
- Se optimiza el indice
- Se almacena el archivo digital en el repositorio de archivos

Flujo de Eventos



Especificación

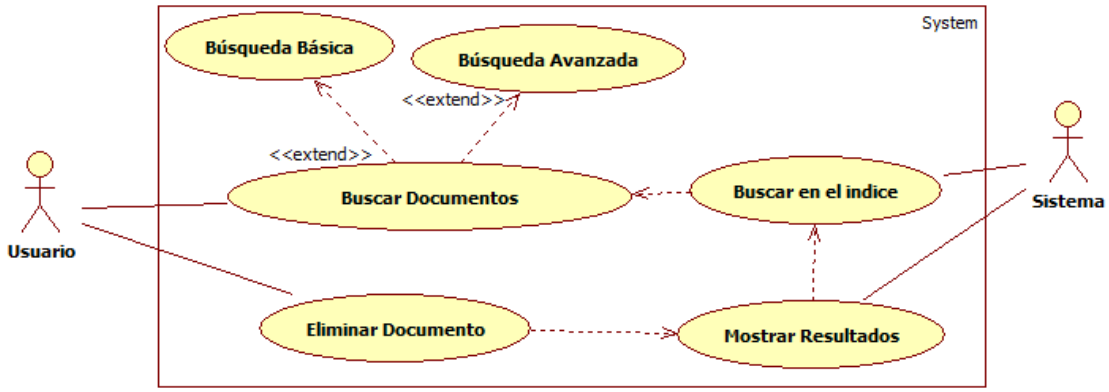
Actividad	**Descripción**
Editar Documento	El usuario selecciona un archivo a editar. Este proceso se realiza en la interfaz.
Editar Datos	Se recuperan los datos del documento a editar, se cargan en la interfaz y se permiten editar, eliminar o agregar metadatos. En la clase RepositoryLucene se recupera el documento por medio de Retrieve, el cual trae un documento por medio del identificador.
Eliminar Documento del Índice	Se elimina físicamente el documento del índice. El método delete de la clase Repository Lucene, el cual recibe el identificador del documento.
Optimizar Indice	Se optimiza el índice para que no queden registros vacios y se presente error o fragmentar el índice. Clase RepositoryManager método optimizeIndexer.
Proceso de Indexación	Es el proceso de indexación cuando se crear un documento.

BÚSQUEDA.

La calidad de una búsqueda en un índice depende de la precisión y la eficiencia en los resultados.

Lucene se encarga de leer e interpretar el query ingresado por el usuario y buscar en el índice sus coincidencias. Por defecto las presenta en un vector de resultados ordenadas por ranking o numero de coincidencias en un mismo documento.

Diagrama de Casos de uso



Buscar Documentos-Buscar en el Índice

Descripción:

Este caso de uso se encarga de recuperar los documentos almacenados e indexados según los criterios de búsqueda ingresados por el usuario.

Para lograr esto, Lucene cuenta con las herramientas necesarias para descomponer y analizar el query de búsqueda y retornar los resultados con más precisión.

La búsqueda se puede realizar de 2 formas:

- Búsqueda sencilla: Consiste en ingresar las palabras a buscar y el sistema examinará en todos sus documentos indexados en cada uno de los campos si existe alguna coincidencia con dichos criterios. Es decir, a Lucene no le importará en que campo se debe buscar la palabra sino que buscará en todo el repositorio.
- Búsqueda avanzada: Le permite al usuario seleccionar en que campos específicos se debe buscar la palabra. Puede buscar en uno o mas campos y generar condiciones lógicas “Y” y “O” entre ellos.

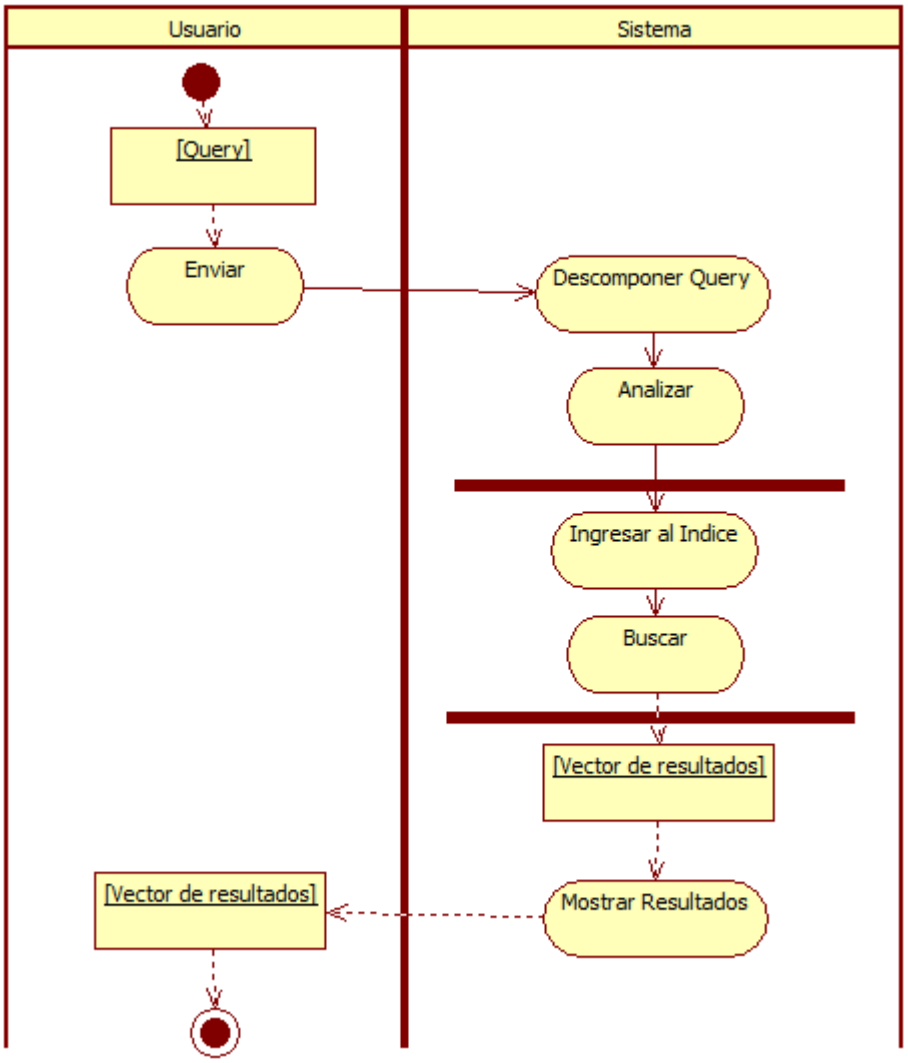
Precondiciones:

- El usuario debe ingresar un query de búsqueda legible para Lucene.
- Debe existir un repositorio configurado en el configuration.properties.

Postcondiciones:

- El sistema retorna los documentos coincidentes con los parámetros ingresados por el usuario.

Flujo de Eventos



Especificación

Actividad	**Descripción**
Enviar Query	El usuario escribe el query y lo envía al servidor. Esto se realiza en la interfaz.
Descomponer Query	Lucene recibe el query como una cadena de caracteres o como un vector de campos con sus datos asocidos. La clase RepositoryLucene se encarga de descomponerla por medio del método query.
Analizar	Luego el query es analizado. Dentro de la clase RepositoryLucene con el método Query.
Ingresar al índice	El sistema abre y lee el índice. Con el método startReader de la clase RepositoryManager
Buscar en el índice	El sistema busca en el índice el query retornado por el Analizer. Para esto de debe buscar en el índice por medio del metodo startSearcher llamado desde el método Query de la clase RepositoryManager
Mostrar Resultados	El sistema almacena los resultados en hits.

Eliminar Documento

Descripción:

Luego de que el sistema retorne los resultados de la búsqueda, el sistema le permite al usuario eliminar cualquier documento del índice.

Si el documento tiene archivo digital asociado, éste también es eliminado del repositorio de archivos.

Cada que se elimina un archivo, el sistema se encarga de optimizar el índice.

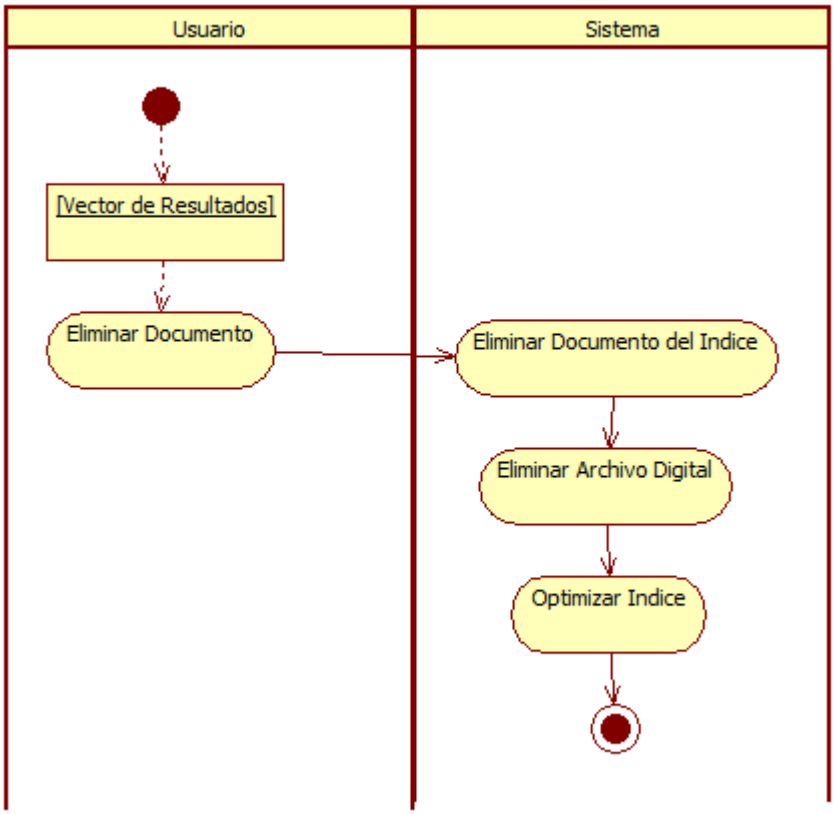
Precondiciones:

- Seleccionar un documento a eliminar luego del resultado de una búsqueda.
- El documento existe en el índice

Postcondiciones:

- Se eliminan tanto los metadatos como el contenido del archivo.
- Se elimina el archivo digital del repositorio de archivos.
- Se optimiza el índice.

Flujo de Eventos



Especificación

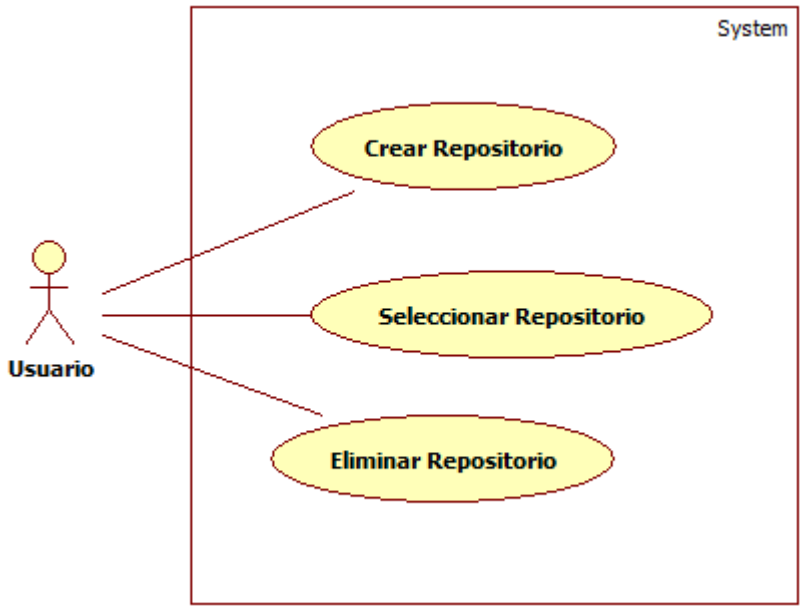
Actividad	**Descripción**
Eliminar Documento	El usuario se encarga de seleccionar un documento para eliminar. Desde la interfaz
Eliminar Documento del Índice	El sistema elimina el documento del índice. Desde la clase RepositoryLucene por medio del método delete
Eliminar Archivo Digital	El sistema elimina el archivo digital asociado al documento del repositorio de archivos indexados. El método delete se encarga de realizar dichas verificaciones.

Optimizar índice	El sistema optimiza el índice. El método optimizeIndexer de RepositoryManager.

ADMINISTRAR REPOSITORIOS.

El sistema permite crear varios repositorios para los índices. Además permite eliminar y seleccionar el índice en el cual se desea buscar o agregar un documento.

Diagrama de Casos de uso



Especificación

Actividad	**Descripción**
Crear Repositorio	El usuario puede crear un nuevo índice. Cada que se crea, Lucene agrega un nuevo archivo al repositorio de índices configurado en index_dir en el archivo configuration.properties
Seleccionar Repositorio	EL usuario selecciona el repositorio en el cual desea buscar o agregar un nuevo documento. El sistema despliega los índices ya creados y el seleccionado se configurará en el configuration.properties en la propiedad index_dir.
Eliminar Repositorio	El usuario selecciona eliminar un repositorio. Lucene elimina los archivos que se crearon cuando se agrego el índice.

ANEXOS

Cambios en el BDNG Core

- Se agregó la varible “filename”, y los metodos getFileName y setFileName a la clase RecordDC para almacenar la ruta del archivo asociado al documento.
- Se agregaron los metodos getRank y setRank en la clase Record para traer el ranking del registro según los criterios de búsqueda.
- Se agregó el metodo getNumElements que trae el numero de elementos que tiene un modelo de metadatos.