UNIVERSIDAD POLITÉCNICA DE PUEBLA

Maestría en Ingeniería

"INTERFAZ DE VISUALIZACIÓN JERÁRQUICA PARA COLECCIONES DE DOCUMENTOS"

TESIS DE MAESTRÍA

DULCE KARINA TREJO TLAPA

Juan C. Bonilla, Puebla. Octubre 2013

UNIVERSIDAD POLITÉCNICA DE PUEBLA

Maestría en Ingeniería

"INTERFAZ DE VISUALIZACIÓN JERÁRQUICA PARA COLECCIONES DE DOCUMENTOS"

TESIS DE MAESTRÍA

DULCE KARINA TREJO TLAPA

COMITÉ EVALUADOR

DRA. MARIA AUXILIO MEDINA NIETO

ASESOR

DRA. ERIKA ANABEL MARTÍNEZ MIRÓN

SINODAL

DR. JORGE DE LA CALLEJA MORA

SINODAL

Juan C. Bonilla, Puebla. Octubre 2013

Índice

Capítulo 1

Planteamiento del problema de investigación

1.1 Introducción

Internet ha impulsado la construcción de nuevas formas de comunicación y de compartir el conocimiento. Las bibliotecas digitales son un ejemplo de ello ya que permiten tener acceso, organizar, mantener, compartir y preservar colecciones digitales de documentos con información validada. Los documentos generalmente están clasificados y agrupados en una o más categorías, sin embargo, la cantidad de éstos puede ser muy grande, lo cual podría dificultar su localización.

El uso de técnicas de búsqueda se estudia en recuperación de información y un factor determinante que permite obtener un mayor o menor aprovechamiento de los datos lo es la presentación de resultados. Las listas ordenadas de referencias tradicionales pueden perder su eficacia cuando el número de documentos obtenidos es elevado.

La utilización de técnicas visuales, ya sea en dos o tres dimensiones, muestran de una sola vez conjuntos de datos de forma organizada en una vista común fácil de navegar y de entender por los usuarios. El objetivo de agrupar es resumir datos de un tamaño elevado, lo cual implica ordenar, clasificar y expresar en términos de frecuencia de clases cómo ha sido organizada la información.

En esta tesis se aborda la aplicación de técnicas visuales a colecciones de documentos que pueden estar clasificados o no; se describe la implementación de un sistema de visualización de documentos para colecciones digitales organizadas jerárquicamente, así como una alternativa de tratamiento para documentos que no están clasificados que se basa en el uso de los algoritmos COBWEB y K-Medias implementados en Weka.

1.2 Objetivo general

Construir un mecanismo de visualización personalizado para documentos digitales organizados jerárquicamente.

1.3 Objetivos específicos

- ✓ Analizar técnicas de agrupamiento aplicadas a documentos.
- ✓ Diseñar un esquema de representación personalizado de la información y su organización jerárquica.
- ✓ Diseñar una interfaz de visualización de documentos flexible y personalizado.
- ✓ Implementar técnicas de búsqueda de información para bibliotecas digitales en la interfaz de visualización.

1.4 Justificación

La sobrecarga de información que producen los sistemas de recuperación de información cuando proporcionan listas de resultados, puede llegar a invalidar su utilidad, debido a que se dificulta la consulta, además, puede provocar desorientación en el usuario.

La consulta de una colección enfrenta retos para los desarrolladores ante la constante problemática de los usuarios al no encontrar lo que requieren. Una alternativa de solución es el diseño de vistas significativas que permitan reconocer fácilmente patrones, creando interfaces comprensibles que sirvan de apoyo para especificar qué es lo que quiere buscar y proporcionar una visualización eficiente de los resultados. Por tanto, haciendo uso de técnicas de visualización, es posible representar relaciones de similitud así como resaltar información.

En general, la clasificación y el agrupamiento jerárquico de documentos permiten mantener organizada la información de forma automática, lo cual es adecuado para colecciones grandes.

Un sistema de visualización facilita la navegación dentro de colecciones y permite una mejor comprensión de la información, emplea ambientes de representación en 2D o 3D [1].

Las interfaces visuales se han hecho familiares a los usuarios en aplicaciones informáticas numerosas, debido a que una representación visual puede comunicar algunos tipos de información de forma más rápida y eficaz; así, se busca comunicar un concepto abstracto de una manera más familiar y accesible mediante el empleo de metáforas [2].

En esta tesis se aplican conceptos de visualización de información y usabilidad en el desarrollo de una interfaz dirigida a usuarios con familiaridad en el uso de bibliotecas digitales y/o en tareas de consulta de colecciones. Se ha desarrollado una interfaz que pretende ser sencilla, intuitiva, dinámica, personalizada y comprensible, la cual podría apoyar la localización de documentos de interés, así como también mostrar las relaciones existentes entre las clases de los documentos. En la tesis se considera que cada documento está descrito en un registro, término que hace referencia a un conjunto de metadatos en el protocolo "Open Archives Initiative Protocol for Metadata Harvesting" (OAI-PMH).

1.5 Contribuciones

- Representación jerárquica de registros del protocolo OAI-PMH.
- Interfaz de visualización personalizada de una colección de documentos en un ambiente 3D.
- Aplicación de algoritmos de agrupamiento jerárquicos a colecciones de documentos.
- Recuperación de información asociada a la visualización de colecciones.

1.6 Recursos

El entorno para el desarrollo de un proyecto incorpora hardware y software. El hardware proporciona una plataforma para soportar las herramientas de software

utilizadas para desarrollar el producto de trabajo. Las secciones siguientes describen estos recursos.

1.6.1 Hardware

Para el desarrollo de esta tesis se utiliza una computadora portátil MacBook Pro con las siguientes características:

- Procesador Dual Core de 2.3 Ghz, Intel Core i5
- 4 Gb de memoria DDR3 de 1600 MHz
- 500 Gb de disco duro (5400 rpm).

1.6.2 Software

- ***JDK.*** Es el acrónimo de "Java Development Kit", es decir, kit de desarrollo de Java. Se puede definir como un conjunto de herramientas, utilidades, documentación y ejemplos para desarrollar aplicaciones Java [3].

  En esta tesis se usa la versión 1.7 de JDK para implementar la interfaz de visualización y un componente para establecer la comunicación con Weka.

- ***NetBeans.*** Es un entorno de desarrollo integrado libre, hecho principalmente para el lenguaje de programación Java, permite que las aplicaciones sean desarrolladas a partir de un conjunto de componentes de software llamados *módulos* [4]. Se usó la versión 7.3.

- ***El API Java 3D.*** Es una interfaz para escribir programas que muestran e interactúan con gráficos tridimensionales [3]. Para la creación de la interfaz se usó la versión 1.5.1.

- ***Weka.*** Es un conjunto de librerías Java diseñadas para la extracción de conocimiento desde bases de datos desarrollado en la Universidad de Waikato bajo licencia GPL [5]. La versión empleada fue la 3.6.9.

La interfaz de visualización se desarrolla como una aplicación local que puede personalizarse de acuerdo a los intereses y preferencias de los usuarios.

Capítulo 2

Marco teórico

En la actualidad, es notable el crecimiento de los volúmenes de la información en la red, si además se agrega que es poco probable que ésta sea confiable o validada, entonces es notoria la necesidad de proponer alternativas de solución a problemas de veracidad, organización y representación de la información.

Las bibliotecas digitales son una forma de acceder a fuentes con cierto grado de credibilidad, son recursos alojados en la red donde los usuarios pueden hacer uso de servicios que apoyan el almacenamiento, preservación y consulta de documentos organizados en colecciones.

Un problema al que se puede enfrentar una biblioteca digital es la organización, es decir, que los documentos se encuentran validados pero no organizados, una forma conocida de abordarlo es hacer uso de herramientas que a través de la aplicación de algoritmos de agrupamiento, proporcionen como salida la organización de documentos para representarse posteriormente en algún formato que facilite la búsqueda y recuperación.

Por lo anterior, representar la información contenida en una colección conlleva al desarrollo de interfaces que permitan visualizar documentos de forma tal que sea sencilla o comprender su organización. El uso de técnicas de visualización de información (VI) juega un papel muy importante ya que permitiría cumplir con dicho objetivo.

La Figura 1 muestra las tareas principales propuestas para visualizar documentos en colecciones cuando los registros se encuentren: 1) clasificados o 2) no clasificados.

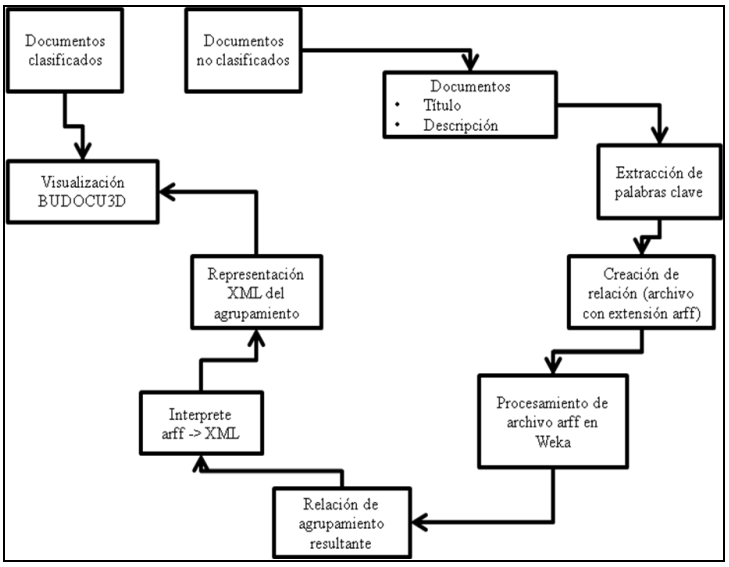


Figura 1 Relaciones entre tareas para implementar la interfaz.

La interfaz permite visualizar documentos de una colección, los datos de entrada principales corresponde a un conjunto de registros OAI-PMH. Físicamente, la entrada es una carpeta de archivos con extensión XML, uno para cada registro.

El proceso comienza definiendo el tipo de colección; cuando ésta esté clasificada jerárquicamente, entonces se puede visualizar de forma directa al utilizar una representación en XML. En el caso en que los registros no estén clasificados, entonces se realiza un proceso de limpieza de datos para extraer palabras clave del título y la descripción, se crea un archivo que guarda estas palabras para posteriormente procesarlas con algún algoritmo de agrupamiento en Weka y así obtener los resultados e interpretarlos de tal manera que el resultado final del proceso sea obtener una representación en XML que sirva de entrada para la interfaz de visualización.

La sección siguiente contiene los conceptos básicos para comprender este caso.

2.1 Agrupamiento

La minería de datos representa la posibilidad de extraer información nueva y útil, el agrupamiento es una de las principales tareas en el proceso de minería de datos

utilizado para descubrir grupos e identificar distribuciones y características interesantes en los datos.

El proceso de agrupar un conjunto de objetos físicos o abstractos dentro de clases con objetos similares se denomina clustering o *agrupamiento*, que consiste en agrupar una colección dada de datos no etiquetados en un conjunto de grupos de tal manera que los objetos que pertenecen a un grupo sean homogéneos entre sí, buscando además que la heterogeneidad entre los grupos distintos sea lo más elevada posible [6].

El agrupamiento de datos es una tarea de análisis exploratorio que se refiere a la clasificación de patrones de una forma no supervisada, formando grupos en base a las relaciones no perceptibles a simple vista, con el objetivo de descubrir una estructura subyacente.

Se define un patrón como lo opuesto al caos, una entidad vagamente definida que puede ser nombrada. La utilidad en el análisis exploratorio de datos está demostrada por su uso en diversos contextos y disciplinas como la recuperación de información, la minería de datos o la segmentación de imágenes, entre otras [7].

2.1.1 Objetivo del agrupamiento

El objetivo del agrupamiento es organizar un conjunto de objetos en grupos, de forma tal que los objetos dentro de un grupo posean un alto grado de semejanza, mientras que los pertenecientes a grupos diferentes sean poco semejantes entre sí.

Una característica relevante del agrupamiento respecto a la clasificación es que son precisamente los grupos y la pertenencia a los grupos lo que se quiere determinar *a priori*, es decir, sin saber cómo son los grupos ni cuantas o cuales categorías existen.

En agrupamiento de datos, la clasificación de patrones se da de forma no supervisada, que a diferencia de la clasificación supervisada en la que se tienen

patrones previamente clasificados, el objetivo radica en agrupar el conjunto de patrones no etiquetados en agrupamientos con algún significado [7], en otras palabras, los grupos que hayan sido generados deben tener sentido y una relación existente para determinar que son de utilidad.

2.1.2 Clasificación de técnicas de agrupamiento

El agrupamiento puede considerarse de diferentes maneras dependiendo del algoritmo seleccionado y de sus propiedades. Los grupos de salida (clusters) pueden ser *rígidos* o *difusos*, el primero de ellos realiza una partición de los datos en grupos y en el segundo, cada patrón tiene un grado variable de la calidad en cada uno de los grupos de salida, implica la posibilidad de que un dato se asocia a dos o más grupos.

Considerando al agrupamiento de datos como una tarea de clasificación no supervisada, los métodos pueden dividirse en *paramétricos* y no *paramétricos* [8].

Entre los métodos de agrupamiento paramétricos se encuentran las mixturas finitas, éstas son una herramienta para modelar densidades de probabilidad de conjuntos de datos univariados y multivariados, modelan observaciones que se asume han sido producidas por un conjunto de fuentes aleatorias alternativas e infieren los parámetros de estas fuentes para identificar qué fuente produjo cada observación, lo que lleva a un agrupamiento del conjunto de observaciones.

Los métodos de agrupamiento no paramétricos pueden dividirse en tres grupos: *jerárquicos, particionales y basados en densidad*. Los algoritmos jerárquicos son aquellos en los que se va particionando el conjunto de datos por niveles, de modo tal que en cada nivel generalmente, se unen o se dividen dos grupos del nivel anterior, según si es un algoritmo *aglomerativo o divisivo*.

Los algoritmos particionales son los que realizan una división inicial de los datos en grupos y luego mueven los objetos de un grupo a otro según se optimice alguna función objetivo.

Los algoritmos basados en densidad enfocan el problema de la división de una base de datos en grupos teniendo en cuenta la distribución de densidad de los puntos, de modo tal que los grupos que se forman tienen una alta densidad de puntos en su interior, mientras que entre ellos aparecen zonas de baja densidad. La sección siguiente contiene algunas de las características del agrupamiento jerárquico.

2.1.3 Agrupamiento jerárquico

El dendograma es la representación gráfica que ayuda a interpretar el resultado de un análisis de grupos puesto que queda reflejada la formación de los grupos, así como las distancias entre ellos se asocia a la forma de medir su similitud [9]. En la Figura 2 se muestra un ejemplo de un dendograma.

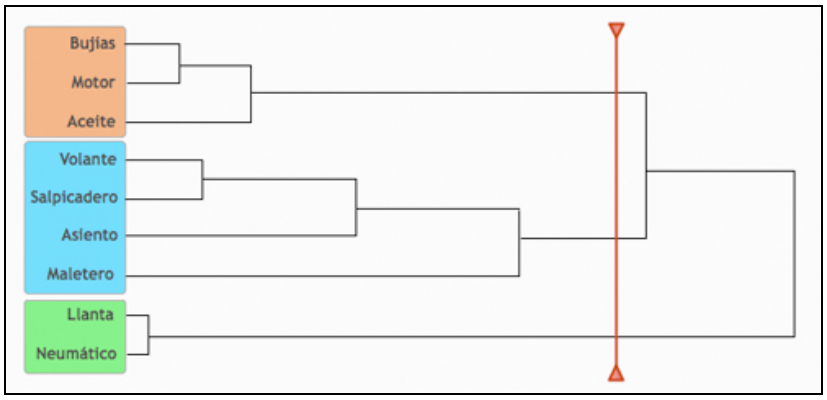


Figura 2 Ejemplo de dendograma

Así, como se puede ver en la Figura 2, los términos que son más cercanos entre sí, se agrupan en conjuntos de colores. Esto significaría que, cada uno de éstos grupos de términos podrían formar parte de una misma sección o subsección. La línea vertical indica el límite para crear las agrupaciones.

Según la forma de construcción del dendograma, los métodos jerárquicos pueden clasificarse a su vez en *aglomerativos y divisivos*. Los métodos aglomerativos construyen la jerarquía de abajo hacia arriba, creando un grupo por objeto, para luego unirlos gradualmente hasta que todos los objetos pertenezcan al mismo grupo. Los métodos divisivos construyen la jerarquía de arriba hacia abajo,

creando inicialmente un único grupo al que pertenecen todos los objetos para luego ser dividido gradualmente [10].

Después de haber presentado cómo se clasifican las técnicas de agrupamiento, es posible mencionar algunos algoritmos utilizados frecuentemente, en esta tesis se describen COBWEB (jerárquico) y K-Medias (particional), de los cuales se amplía su información en secciones posteriores.

Existe diversidad de software para realizar el análisis de grupos por ejemplo Cluster 3.0, el cual proporciona una interfaz gráfica de usuario que permite acceder a las técnicas de agrupamiento, está disponible para Windows, Mac OS X y Linux / Unix.

Para ver los resultados de la agrupación generados por Cluster 3.0, se recomienda utilizar TreeView Java Alok Saldanha, que muestra tanto agrupamientos jerárquicos como particionales. Otra de las herramientas existentes es Pycluster 1.3 que al igual que Cluster 3.0, proporciona rutinas que pueden ser ancladas al lenguaje C [11].

En esta tesis se emplea Weka, que es una herramienta creada por la Universidad de Waikato en la cual se cuenta con un apartado exclusivo para realizar análisis de algoritmos de agrupamiento ya implementados en su interfaz. La siguiente sección presenta una breve introducción a Weka y las características principales de su interfaz y su funcionalidad.

2.2 Entorno Weka

WEKA [5] (*Waikato Environment for Knowledge Analysis)* es una herramienta que permite la experimentación de análisis de datos mediante la aplicación y evaluación de técnicas provenientes del aprendizaje automático. **E**s un programa de distribución y difusión libre, programado en Java e independiente de la arquitectura debido a que funciona en cualquier plataforma sobre la que haya una máquina virtual disponible.

Es un conjunto de librerías que apoyan la minería de datos y pueden ser llamadas desde la interfaz de Weka o desde clases Java propias. Contiene herramientas para diferentes tareas básicas como las siguientes:

- **Pre procesos**: Herramientas para el pre procesamiento de los datos, por ejemplo, la discretización de variables.
- **Clasificación**: Algoritmos de clasificación como ID3 o C4.5
- **Cluster**: Algoritmos de agrupamiento como K-Medias simple, COBWEB.
- **Asociación**: Algoritmos para encontrar relaciones de asociación entre variables *a priori*.
- **Selección de atributos**: Una vez cargados los datos, Weka es capaz de buscar por el usuario las mejores variables del modelo. Un modelo pretende encontrar relaciones y patrones de comportamiento en el conjunto de datos para ofrecer conocimiento nuevo sobre un problema.
- **Visualizar**: Herramienta de visualización de datos en los ejes cartesianos.

2.2.1 Interfaz de WEKA

La interfaz gráfica de Weka cuenta con cuatro formas de acceso a las diferentes funcionalidades de la aplicación.

- ***Simple CLI*** *(Simple command-line interface):* acceso a través de la consola de comandos a todas las opciones de Weka.

- ***Explorer***: acceso a las principales características del programa.

- ***Experimenter****:* permite la comparación sistemática de una ejecución de los algoritmos predictivos de Weka sobre una colección de conjuntos de datos.

- ***Knowledge Flow**:* soporta esencialmente las mismas opciones que la interfaz Explorer, pero ésta permite "arrastrar y soltar". Ofrece soporte para el aprendizaje incremental.

En el desarrollo de esta tesis se trabaja con la funcionalidad de ***Explorer,*** ya que, es en este apartado se encuentran las herramientas para realizar agrupamiento y procesar los datos.

2.2.2 Preparación de los datos para procesarlos en Weka

Para poder realizar un agrupamiento será necesario procesar los datos para que los resultados que se obtengan sean lo más precisos posibles, es decir, con la menor media de error este proceso de preparación de los datos se le conoce como *caracterización de la información* y se obtiene como resultado un archivo en formato ***arff*** que contiene la información de los datos tales como el nombre, tipo de cada atributo, descripción del origen de los datos, entre otros. ***Arff*** es el formato de los datos de entrada que requiere Weka.

La Figura 3 ilustra un ejemplo de un archivo antes de realizar el proceso de limpieza de los datos.

```xml
<?xml version="1.0" encoding="iso-8859-1"?>
<OAI-PMH xmlns="http://www.openarchives.org/OAI/2.0/" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.openarchives.org/OAI/2.0/ http://www.openarchives.org/OAI/2.0/OAI-PMH.xsd">
<responseDate>2012-06-13T19:20:47Z</responseDate>
<request identifier="http://172.16.30.170/~oai_uppuebla/index.html" metadataPrefix="oai_dc" verb="GetRecord">
</request>
<GetRecord>
<record>
<header>
<identifier>castellanosMejia.pdf</identifier>
<datestamp>2012-06-13T11:13:36Z</datestamp>
</header>
<metadata>
<oai_dc:dc xmlns:oai_dc="http://www.openarchives.org/OAI/2.0/oai_dc/" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:dc="http://purl.org/dc/elements/1.1/" xsi:schemaLocation="http://www.openarchives.org/OAI/2.0/oai_dc/
http://www.openarchives.org/OAI/2.0/oai_dc.xsd">
<dc:coverage>Licenciatura</dc:coverage>
<dc:title> Desarrollo de una aplicacion web para calendarizar eventos turisticos municipios del estado de Puebla </dc:title>
<dc:publisher>Universidad Politecnica de Puebla</dc:publisher>
<dc:creator>Daniel Castellanos Mejia </dc:creator>
<dc:contributor>No disponible</dc:contributor>
<dc:contributor>No disponible</dc:contributor>
<dc:language>es</dc:language>
<dc:type>Technical report</dc:type>
<dc:type>Reporte tï¿½cnico</dc:type>

<dc:subject>K. Aplicaciones Web</dc:subject>
<dc:subject>K.2 APLICACIONES WEB</dc:subject>

<dc:date>2003-05-09</dc:date>
<dc:identifier>
172.16.30.170/~oai_uppuebla/rt/castellanosMejia.pdf
</dc:identifier>
<dc:description>
El presente proyecto de investigacion expone una alternativa de apoyo a la gestion de los datos de los tutorados de
Ingenieria en Informatica de la UPPuebla, mediante la propuesta
de construccion dinamica de reportes graficos sobre el desempeno academico de tutorados, desarrollada con el lenguaje de
programacion PHP en conjunto con la herramienta FushionCharts, con el objetivo de obtener de manera dinamica reportes en
forma tabular y grafica sobre el
desempeno de los tutorados en la UPPuebla.
</dc:description>
<dc:format>application/pdf</dc:format>
</oai_dc:dc>
</metadata>
</record>
</GetRecord>
</OAI-PMH>
```

Figura 3 Ejemplo de un registro del protocolo OAI-PMH

En la figura 4 es posible visualizar como queda definido un archivo arff, en el cual se almacenan las características de los documentos definidos por una relación, donde cada palabra clave obtenida es un atributo y la definición de datos corresponde a la frecuencia de repetición de los atributos.

```
@RELATION grupouno150_Cv2
@ATTRIBUTE ensenanza NUMERIC          %50%
@ATTRIBUTE notas NUMERIC              %44%
@ATTRIBUTE caso NUMERIC               %40%
@ATTRIBUTE galaxias NUMERIC           %21%
@ATTRIBUTE traves NUMERIC             %17%
@ATTRIBUTE desarrollo NUMERIC         %16%
@data
0,0,0,0,0,0,0,0,18,18,16,16,0,0,0,0,0,0,0,0,0,10,0,0,0,0,9,0,0,0
,0,0,0,0,0,0,8,8,8,0,0,0,0,7,0,0,0,0,0,0,0,0,0,0,6,0,0,0,0,0,6
,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
,0,0,0,0,0,0,0,0,4,4,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,4,0,0,4,4,4
,4,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
,0,0,0,0,0,0,0,0,0,3,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,2,2,2,2,2,2,2,2,2,2,2
,2,2,2,2,2,2,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
,0,0,0,0,0,0,0,0,0,0,0
%/Users/KryTrejo/InfoKry's/OTROSDES/pruebasweka/plantillaRegistro1.xml
```

Figura 4 Ejemplo de caracterización de la información

2.2.3 Algoritmos de agrupamiento en WEKA

El proceso de agrupamiento de una serie de documentos según criterios habitualmente de distancia, es soportado por Weka través de la aplicación de algoritmos de agrupamiento incluidos en su interfaz. A continuación se describen algunos de ellos y sus características principales.

2.2.3.1 COBWEB

COBWEB es un algoritmo de agrupamiento jerárquico que se caracteriza porque utiliza aprendizaje incremental, esto es, realiza las agrupaciones instancia a instancia. Durante la ejecución del algoritmo se forma un árbol donde las hojas representan los segmentos y el nodo raíz engloba por completo el conjunto de datos de entrada. Al principio, el árbol consiste en un único nodo raíz. Las instancias se van añadiendo una a una y el árbol se va actualizando en cada paso.

La actualización consiste en encontrar el mejor sitio donde incluir la nueva instancia, operación que puede necesitar de la reestructuración de todo el árbol o simplemente la inclusión de la instancia en un nodo que ya existía.

La clave para saber cómo y dónde se debe actualizar el árbol la proporciona una medida denominada *utilidad de categoría*, que mide la calidad general de una partición de instancias en un segmento. La reestructuración que mayor utilidad de categoría proporcione es la que se adopta en ese paso. El algoritmo es muy sensible a otros dos parámetros:

- ***Acuity***: representa la medida de error de un nodo con una sola instancia, es decir, establece la varianza mínima de un atributo.

- ***Cut-off***: indica el grado de mejoría que se debe producir en la utilidad de categoría para que la instancia sea tomada en cuenta de manera individual.

COBWEB crea un árbol de clasificación, donde cada nodo es un concepto que tiene una descripción probabilística de ese concepto que resume los objetos clasificados bajo ese nodo. La descripción probabilística incluye la probabilidad del concepto P(Ci) y las probabilidades condicionales de pares atributo-valor dado el concepto P(Ai = Vij|Ck). COBWEB utiliza una medida llamada utilidad de la categoría para construir el árbol:

$$CU = \frac{\sum_{k=1}^{n} P(C_k)\left[\sum_i \sum_j P(A_i = V_{ij}|C_k)^2 - \sum_i \sum_j P(A_i = V_{ij})^2\right]}{n} \qquad \textbf{EQ. 1}$$

donde: *n* es el número de clases en un nivel del árbol. La utilidad de la categoría mide el valor esperado de valores de atributos que pueden ser adivinados a partir de la partición sobre los valores que se pueden adivinar sin esa partición. Si la partición no ayuda en esto, entonces no es una buena partición.

COBWEB desciende el árbol buscando el mejor lugar o nodo para cada objeto. Esto se basa en poner el objeto en cada nodo y en un nodo nuevo y medir en cual se tiene la mayor ganancia de utilidad de categoría.

COBWEB también considera en cada iteración unir los dos mejores nodos evaluados y dividir el mejor nodo evaluado. Esto es, cada vez que se selecciona un lugar en un nivel para un nuevo objeto, se consideran los dos mejores objetos (de mayor utilidad) y se determina juntarlos. El caso contrario, sucede una vez que se encuentra el mejor lugar para un objeto nuevo, pero el unir nodos no resulta beneficioso, entonces se considera dividir ese nodo. COBWEB depende del orden de los objetos. La división entre el número de grupos sirve para incentivar a tener grupos con más de un elemento. COBWEB asume que la distribución de probabilidad de los atributos es independiente de las demás. El algoritmo se puede extender a valores numéricos usando distribuciones gaussianas.

$$f(a) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(a-\mu)^2}{2\sigma^2}}$$ **EQ.2**

El equivalente a la sumatoria de probabilidades es:

$$\sum_j P(A_i = V_{ij})^2 \sim \int f(a_i)^2 da_i = \frac{1}{2\sqrt{\pi}\sigma_i}$$ **EQ.3**

Se estima la desviación estándar del atributo numérico con los datos en el grupo y en los datos para todos los grupos:

$$CU = \frac{1}{k} \sum_{k=1}^{n} P(C_k) \frac{1}{2\sqrt{\pi}} \sum_i \left(\frac{1}{\sigma_{ik}} - \frac{1}{\sigma_i} \right)$$ **EQ.4**

Si la desviación estándar es cero, el valor de utilidad se vuelve infinito, por lo que se impone un valor de varianza mínimo en cada atributo (acuity). El otro parámetro que se usa en COBWEB es el de corte (cutoff), que básicamente se usa para parar la generación de nodos nuevos.

COBWEB pertenece a los métodos de aprendizaje conceptual o basados en modelos, a este algoritmo no hay que proporcionarle un número de grupos [12].

2.2.3.3 K- Medias

El algoritmo K-Medias se trata de un algoritmo clasificado como método de particionado y recolocación; representa a cada uno de los grupos por la media (o media ponderada) de sus puntos, es decir, por su centroide; tiene la ventaja de poseer un significado gráfico y estadístico inmediato. La suma de las discrepancias entre un punto y su centroide, expresado a través de la distancia apropiada, se usa como función objetivo. La función objetivo, suma de los cuadrados de los errores entre los puntos y sus centroides respectivos, es igual a

la varianza total dentro del propio grupo [12].

Existen dos versiones de K-Medias.

- La primera se basa en dos pasos iterativos: primero reasigna todos los puntos a sus centroides más cercanos y en segundo lugar, re-calcula los centroides de los grupos nuevos creados en el paso anterior. El proceso continua hasta alcanzar un criterio de parada.

- La segunda versión reasigna los puntos basándose en un análisis más detallado de los efectos causados sobre la función objetivo, al mover un punto de su grupo a otro nuevo. Si el traslado es positivo, se realiza, en caso contrario, se queda como está.

A diferencia de COBWEB, K-Medias necesita la especificación previa del número de clusters que se desean obtener.

Toma como parámetro *k* que es el número de clusters que forma. Selecciona *k* elementos aleatoriamente, los cuales representan el centro o media de cada grupo. A cada objeto restante se le asigna el grupo con el cual se parece más, basándose en una distancia entre el objeto y la media del cluster. Después calcula la media nueva del grupo e itera hasta no cambiar de medias [12].

selecciona *k* objetos aleatoriamente
repite
re (asigna) cada objeto al grupo más similar con el valor medio
actualiza el valor de las medias de los grupos
hasta
no hay cambio

Normalmente se utiliza una medida de similaridad basada en el error cuadrático:

$$E = \sum_{i=1}^{k} \sum_{p \in C_i} |p - m_i|^2$$

EQ.5

donde: P representa al objeto y m*i* a la media del grupo *Ci* (ambos son objetos multidimensionales). (K-medias) es susceptible a valores extremos porque distorsionan la distribución de los datos. También se pueden utilizar las modas (K-modas) para agrupar objetos categóricos.

Otra posibilidad es usar medianas (K-medianas) para agrupar en base al objeto más representativo del grupo. La idea básica es encontrar un objeto representativo. La estrategia es reemplazar una de las medianas por otro objeto en forma aleatoria y medir si la calidad de los grupos resultantes mejora.

La calidad se evalúa con base a una función de costo que mide la disimilaridad promedio entre un objeto y la mediana en su grupo. Para ver si un objeto aleatorio (*O aleatorio*) es un buen reemplazo de la mediana (*O actual*) se consideran todos los objetos que no sean medianas y se analiza la re-distribución de los objetos a partir de la cual se calcula un costo base, por ejemplo, en el error cuadrático. Esto se repite hasta que no exista mejora, no garantiza encontrar el mínimo global, por lo que se recomienda correr varias veces el algoritmo con diferentes valores iníciales. Otra variante es hacer (K-medias) jerárquico, en donde se empieza con *k=2* y se continua formando grupos sucesivos en cada rama.

La sección siguiente describe los conceptos de visualización de información que se utilizan en la interfaz propuesta para representar jerarquías de documentos.

2.3 Visualización de información (VI)

Ante un crecimiento desmedido de la información, la dificultad de abstracción de características a partir de un alto volumen de datos compartidos crece. Por otra parte, existen técnicas de visualización de información que facilitan la comprensión y abstracción de algunas características.

Visualizar es la formación de imágenes a través de un mapeo visual, auditivo o

táctil de datos en representaciones que pueden ser percibidas. La VI permite la exploración de diversos conjuntos de datos así como la asimilación rápida de información o monitoreo de grandes cantidades de datos [13]. Según [14], la VI es una disciplina transversal que utiliza el poder de comunicación de las imágenes para explicar de manera comprensible las relaciones de significado, causa y dependencia que se pueden encontrar entre las grandes masas abstractas de información que generan los procesos científicos y sociales.

2.3.1 Objetivo de la VI

El objetivo principal de la visualización de datos es transmitir información clara y eficaz a través de medios gráficos para expresar ideas de manera efectiva. Tanto la dimensión estética como la funcional deben ir de la mano y proporcionar determinados puntos de entrada a un conjunto de datos complejos para comunicar sus aspectos clave de forma intuitiva [15].

El diseño adecuado de herramientas de visualización permite simplificar la búsqueda de información, mejora las posibilidades de detección de patrones, aumenta la accesibilidad de los recursos, facilita la utilización de inferencias perceptuales complejas ya implementadas en la corteza visual que permiten comunicar aspectos claves de una forma intuitiva.

2.3.2 Utilización de técnicas de VI

Con el desarrollo de los sistemas de visualización por medio de computadoras, las capacidades provistas por las técnicas de visualización convencionales se han ampliado notablemente, principalmente a través de la posibilidad de interacción directa del usuario con la información visualizada [16].

La Tabla 1 muestra un grupo de técnicas donde es posible conocer el enfoque de investigación, el ambiente de representación, las herramientas desarrolladas, la capacidad de representación y el tipo de visualización que se utiliza. El último

renglón de esta tabla hace referencia a la interfaz propuesta denominada BUDOCU3D (Búsqueda de documentos en cubo 3D).

Tabla 1. Comparativa de técnicas de VI [15].

Técnica	**Enfoque de investigación**	**Ambiente**	**Ejemplos de herramientas**	**Capacidad de visualización**	**Tipo de visualización**
Mapas auto organizados	Redes Neuronales	2D	ET – Mapa WEBSOM	Reducido	Estática
Árboles de extensión mínima	Teoría de grafos	2D	MST CABRO	Reducida	Estática
Vistas hiperbólicas	Teoría de grafos	3D	StarTree HyperTree	Amplia	Dinámica
Mapas jerárquicos	Redes neuronales	3D	Cat a Cone MAPA	Amplia	Dinámica
Mapas de árbol	Teoría de grafos	3D	WebTracer EMTree	Amplia	Dinámica
Ejes jerárquicos	Redes Neuronales	2D	TimeWall TableLens	Reducida	Estática
Agrupamiento jerárquico	Teoría de grafos	3D	BUDOCU3D	Amplia	Dinámica

2.4 API Java 3D, herramienta de VI

La API (*Application Program Interface*) Java 3D es una interfaz de programación de aplicación utilizada para realizar aplicaciones Java y *applets* con gráficos en 3D. Proporciona a los desarrolladores constructores de nivel alto

para crear y manipular geometrías 3D y para implementar las estructuras utilizadas en el renderizado de dichas geometrías.

Esta interfaz se integra con Internet ya que tanto los *applets* como las aplicaciones escritas utilizando Java 3D, tienen acceso al conjunto completo de clases de Java. Los objetos geométricos creados por los constructores residen en un universo virtual que luego es renderizado.

El API está diseñado para crear universos virtuales precisos de una amplia variedad de tamaños. Un programa Java 3D crea objetos Java 3D y los sitúa en un estructura de datos de escenario gráfico. Este escenario es una composición de objetos 3D en una estructura de árbol que especifica completamente el contenido de un universo virtual y cómo va a ser renderizado [3].

La Figura 5 muestra un ejemplo de un objeto generado con el API de java 3D el cual pertenece a la interfaz de visualización BUDOCU3D y corresponde al cubo de clasificaciones para las colecciones de Tesis de Maestría y Reportes Técnicos, también pueden verse los objetos flecha que permiten al usuario ir hacia atrás o hacia adelante en la interfaz.

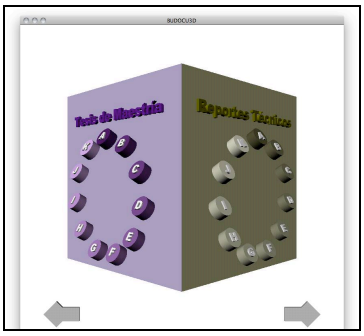


Figura 5 Objeto 3D generado con API Java 3D

2.4.1 Estructura de los objetos en Java 3D

La API de Java 3D está orientada a objetos. Las aplicaciones construyen los elementos gráficos como objetos separados y los conectan unos con otros mediante una estructura en forma de árbol denominada *grafo de escena*. La

aplicación manipula a los objetos utilizando métodos de acceso, modificación y unión definidos en la interfaz se insertan en el grafo (aunque en otra rama distinta), los elementos relacionados con el punto de vista del usuario. Siguiendo dicha estructura desde el nodo raíz hasta los nodos hoja, se ven las distintas operaciones que se realizan para crear la escena final que se quiere conseguir [3].

2.4.2 Aplicaciones y *applets*

Java 3D proporciona la base necesaria para añadir funcionalidades nuevas utilizando código Java. Las aplicaciones de Java 3D pueden incluir objetos definidos utilizando un sistema de CAD o de animación.

Mediante el uso de modeladores externos, se pueden exportar geometrías a archivo, esta información podrá ser utilizada por Java 3D, siempre que la aplicación proporcione un método para leer y traducir dicha información geométrica en primitivas [3]. A manera de ejemplo, la figura 6 muestra cómo es la construcción de un objeto flecha en Java 3D.

```
1  package budocu3d.modelos;
2
3  import com.sun.j3d.utils.geometry.Box;
4  import javax.media.j3d.Appearance;
5  import javax.media.j3d.Geometry;
6  import javax.media.j3d.Transform3D;
7  import javax.media.j3d.TransformGroup;
8  import javax.vecmath.Vector3f;
9
10 public class Flecha3D extends Box
11 {
12 public Flecha3D(Appearance ap,String comando) {
13
14         super(0.05f, 0.03f, 0.005f, ap);
15         for(int i = 0 ; i < 6 ; i++)
16         getShape(i).getGeometry().setCapability(Geometry.ALLOW_INTERSECT);
17
18         TransformGroup tg = new TransformGroup();
19         Transform3D tr = new Transform3D();
20         tr.setTranslation(new Vector3f (0.05f ,0.0f ,0.0f));
21
22         Transform3D ry = new Transform3D();
23         ry.rotZ((Math.PI/4));
24         tr.mul(ry);
25         tg.setTransform(tr);
26
27         Box box2 = new Box(0.04f, 0.04f, 0.005f, ap);
28         box2.setUserData(comando);
29         tg.addChild(box2);
30         for(int i = 0 ; i < 6 ; i++)
31         box2.getShape(i).getGeometry().setCapability(Geometry.ALLOW_INTERSECT);
32         this.addChild(tg);
33         }
34     }
```

Figura 6 Código de objeto flecha en Java 3D

El código que se muestra en la Figura 6 ilustra cómo se realiza la construcción del objeto flecha, el cual extiende a la clase Box que es una clase principal propia del API de Java 3D y se definen sus características generales.

La sección siguiente describe algunas interfaces que han sido desarrolladas para acceder a colecciones de documentos.

2.5 Trabajos relacionados

Las técnicas de agrupamiento y los sistemas bibliotecarios tienen un mismo objetivo: la recuperación y organización temática de la información almacenada.

La combinación de técnicas de agrupamiento y técnicas de VI alcanza un alto poder de representación de la información, potencialmente fácil de comprender y usar sobre colecciones alojadas en bibliotecas digitales.

En esta sección se describen interfaces y sistemas de VI para acceder a colecciones de documentos.

2.5.1 cat – a – cone

Un ejemplo temprano de un sistema de visualización basado en jerarquías de categorías, lo constituye cat-a-cone. El usuario puede ver las etiquetas de las categorías y elegir cualquiera de ellas. Además, puede elegir mayor o menor nivel de detalle y saltar directamente de una categoría a otra. En la Figura 7 se muestra la interfaz de cat-a-cone.

Una característica importante de este sistema, consiste en que la interfaz de visualización permite mostrar simultáneamente, la jerarquía por la que se mueve el usuario y el contenido de cualquier documento que haya querido examinar con detalle [16].

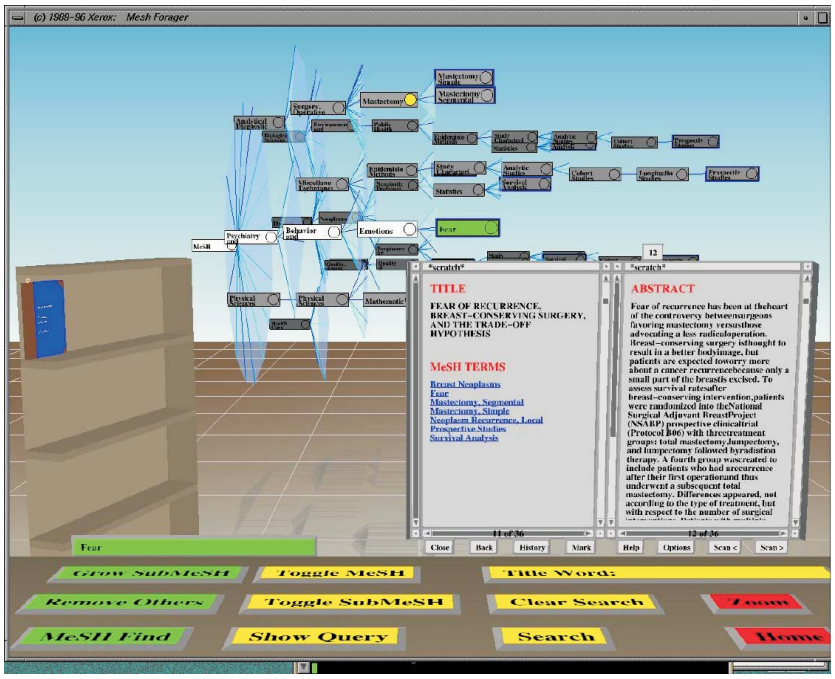


Figura 7 Interfaz del sistema cat-a-cone. [16].

En cat-a-cone, no se visualiza la ruta que ha seguido el usuario para llegar al documento.

2.5.2 GRIDL

«GRIDL - GRaphical Interface for Digital Libraries», es una interfaz de visualización bidimensional con ejes jerárquicos que permite mostrar entre 100 y 10 000 elementos. La profundización en una estructura jerárquica busca un nivel mayor de detalle; sin embargo, puede plantear algún problema de desorientación cuando se aplica simultáneamente a los dos ejes de un espacio bidimensional [16]. La Figura 8 muestra la interfaz del sistema GRIDL.

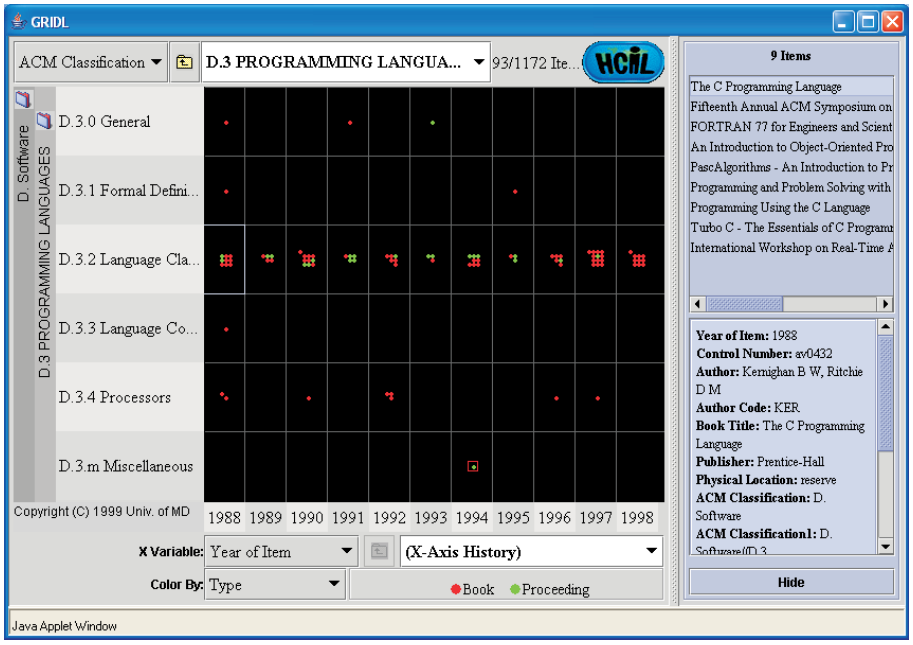


Figura 8 Interfaz del sistema GRIDL [16].

La visualización de las categorías se realiza en el eje y.

2.5.3 SunGroups

SunGroups es un esquema de visualización de información para las publicaciones que se encuentran dentro del espacio que reúne a diferentes colecciones en el proyecto ReMeRi [17]. La visualización consta principalmente de un plano circular dividido en niveles (círculos concéntricos) y sectores (los fragmentos de los círculos concéntricos). Los niveles son los encargados de indicar la jerarquía y los sectores las agrupaciones que se encuentran en cada nivel. La Figura 9 muestra la interfaz principal del sistema SunGroups.

En cada sector se localiza un grupo de palabras clave. El usuario podrá internarse en un sector para encontrar más subgrupos de esas palabras clave, para que al final, obtenga una lista de un número manejable de publicaciones y conocer sus datos como título, institución y año de publicación. Estos elementos sirven de enlace para ir a la ubicación original dentro de las colecciones de ReMeRI y así acceder al documento [18].

Figura 9 Interfaz del sistema SunGroups [18].

Después de conocer el trabajo relacionado, en el siguiente capítulo se aborda el tema de la construcción de un prototipo de interfaz de visualización previo al desarrollo e implementación final, con el propósito de atender algunas de las actividades propuestas en el diseño centrado en el usuario (DCU).

Capítulo 3

Metodología

Se desarrolló una interfaz de visualización que apoya la navegación por colecciones y permite localizar documentos y recuperar información de interés, haciendo uso de técnicas de visualización y aplicando algoritmos de agrupamiento.

La construcción de esta interfaz se hizo siguiendo algunas actividades de la metodología DCU [19] que es una aproximación al diseño de un producto o sistema atento a las necesidades del usuario.

DCU como filosofía de diseño, engloba o se relaciona con un conjunto heterogéneo de metodologías y técnicas que comparten un objetivo común: conocer y comprender las necesidades, limitaciones, comportamiento y características del usuario, involucrando en muchos casos a usuarios potenciales o reales en el proceso. La Figura 10 muestra las fases del diseño centrado en el usuario: 1) Definición de los requerimientos del usuario, 2) Análisis de las necesidades de los usuarios, 3) Desarrollo del producto, 4) Evaluación del proceso.

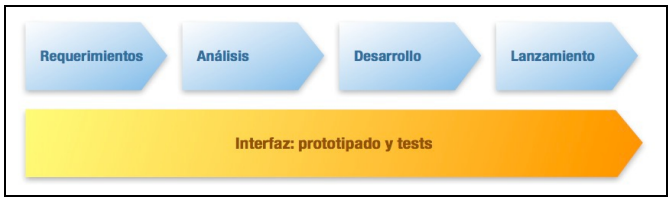


Figura 10 Fases metodológicas del DCU.

Para el desarrollo de la interfaz, se aplicó la metodología como a continuación se enuncia:

Requerimientos: se hizo una revisión por el estado del arte relacionado con la visualización de información obteniendo así los problemas que se enfrentan y las deficiencias existentes en los trabajos relacionados.

Análisis: se hizo un estudio de técnicas de visualización, agrupamiento y recuperación de información para conocer y definir cuáles serán utilizadas en el desarrollo.

Desarrollo: se diseñó un prototipo de interfaz y la implementación en un ambiente3D basado en el prototipo.

Lanzamiento: se puso en uso la interfaz en fase de pruebas.

Interfaz: se realizaron pruebas de usabilidad a la interfaz haciendo uso de cuestionarios donde se reflejan algunas características de usabilidad.

3.1 Prototipo de interfaz del sistema de visualización

El objetivo del diseño del prototipo de interfaz es realizar una verificación de requerimientos de usuario, así como identificar probables errores de secuencia o localización al visualizar colecciones. Centra su diseño en aspectos de visualización y usabilidad, se resaltan características específicas sobre el uso de formas, colores y secuencias para localizar documentos de interés de forma sencilla.

El diseño se realizó haciendo uso de formas cúbicas, circulares, cuadradas y lineales, que son los constructores que provee el API de Java.

El prototipo utiliza valores numéricos para representar el espectro visible de color, es decir, los valores numéricos se ajustaron de acuerdo a las preferencias del usuario.

A cada una de las colecciones y clases se le asocia un número base que se decrementó en intervalos de diez en diez, lo cual permite obtener una gama de color en el mismo rango.

La jerarquía de localización se define durante la navegación, esto permite visualizar con mayor detalle el nivel en el que está localizado el documento de interés.

Como ya se dijo en el Capítulo 2, se denomina BUDOCU3D a la interfaz de visualización. De la Figura 11 a la Figura 14 se indica cómo se realiza el proceso de visualización de un documento en una colección utilizando el prototipo de interfaz.

Paso 1: Ingreso a cubo de colecciones.

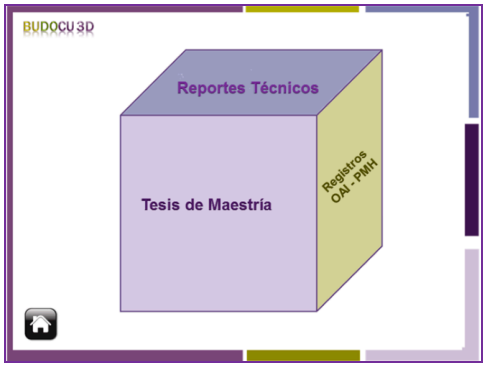


Figura 11 Prototipo de interfaz de VI a colecciones.

Paso 2: Acceso a cubo de clases de una colección.

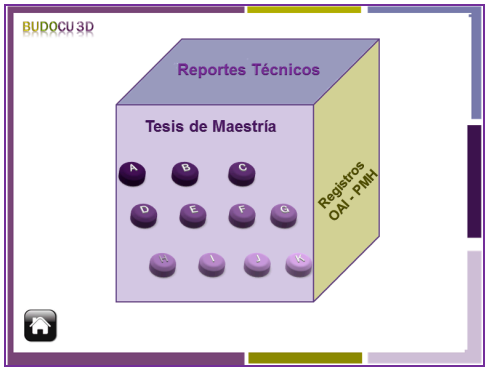


Figura 12 Cubo de clases de una colección

Paso 3: Despliegue de las subclases de una clase de una colección.

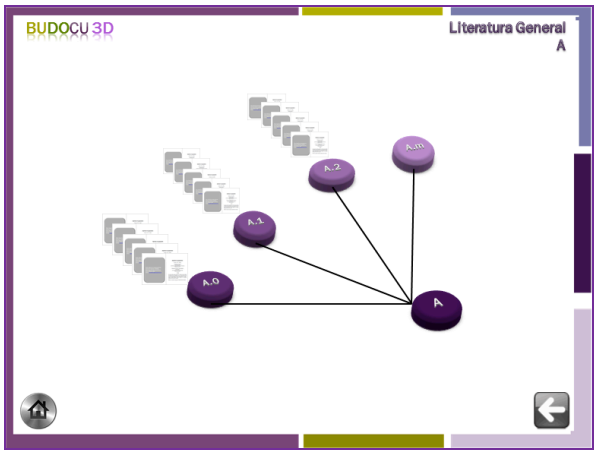


Figura 13 Prototipo de interfaz de VI para acceder a sub clases de una colección.

Paso 4: Visualización de vista previa de documento almacenado en una colección.

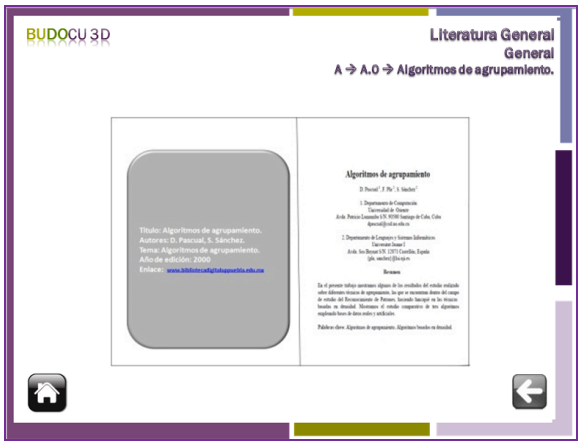


Figura 14 Prototipo de interfaz de VI, vista previa de un documento de una colección.

3.2 Método del análisis de tareas

El análisis de tareas consiste en aprender acerca de las metas de los usuarios, complementa la comprensión debido a que es posible observar directamente las tareas generales que se están tratando de lograr o cómo se realizan actualmente [20].

El análisis de tareas puede ayudar a redefinir tareas que permitan que las tareas coincidan con los usuarios y sus objetivos. La aplicación de métodos que evalúen

el nivel de usabilidad define el éxito de la interfaz, debido a que se han validado satisfactoriamente tanto las necesidades como las facilidades que requiere un usuario para interactuar.

3.2.1 Atributos de usabilidad

La usabilidad es una cualidad abstracta que no puede ser medida directamente. Para poder estudiarla se descompone habitualmente en los siguientes atributos [20]:

Facilidad de aprendizaje: está dado por la facilidad con que los usuarios puedan realizar las tareas en el sistema.

Facilidad de uso: facilidad con la que el usuario hace uso de la herramienta, con menos pasos o más naturales a su formación específica. Tiene que ver con la eficacia y eficiencia de la herramienta.

Satisfacción: medida en que los usuarios están satisfechos con los objetivos logrados.

Eficacia: en su interacción con el sistema, el usuario debe tener un nivel alto de productividad.

Retención sobre el tiempo: se basa en qué tanto puede recordar un usuario la realización de una tarea pasado un tiempo de haberla efectuado.

Satisfacción subjetiva: se basa en qué tan placentera es la utilización del sistema para los usuarios.

Número de errores por parte de los usuarios: la aplicación debe ayudar a que el usuario cometa el menor número de errores posibles mientras esté interactuando.

Tiempo requerido para realizar una tarea: los usuarios interactúan con la aplicación en busca de resultados rápidos, por lo que el tiempo para la respuesta del sistema debe ser de un tiempo relativamente corto.

Flexibilidad: variedad de posibilidades con las que el usuario y el sistema pueden intercambiar información. También abarca la posibilidad de diálogo, la multiplicidad de vías para realizar la tarea y la similitud con tareas anteriores.

Robustez: es el nivel de apoyo al usuario que facilita el cumplimiento de sus objetivos.

Privacidad: el usuario debe confiar en que sus datos personales y actividades sólo van a ser visibles para quien ellos elijan.

Algunos de estos atributos no contribuyen a la usabilidad del sistema en la misma dirección, pudiendo ocurrir que el aumento de uno de ellos tenga como efecto la disminución de otro.

La evaluación de usabilidad permite comprender el mundo de los usuarios, guiar el proceso de diseño y verificar que las necesidades han sido alcanzadas. Una prueba de usabilidad es una técnica formal y su objetivo es estudiar la usabilidad de una aplicación en un entorno real con usuarios reales [20].

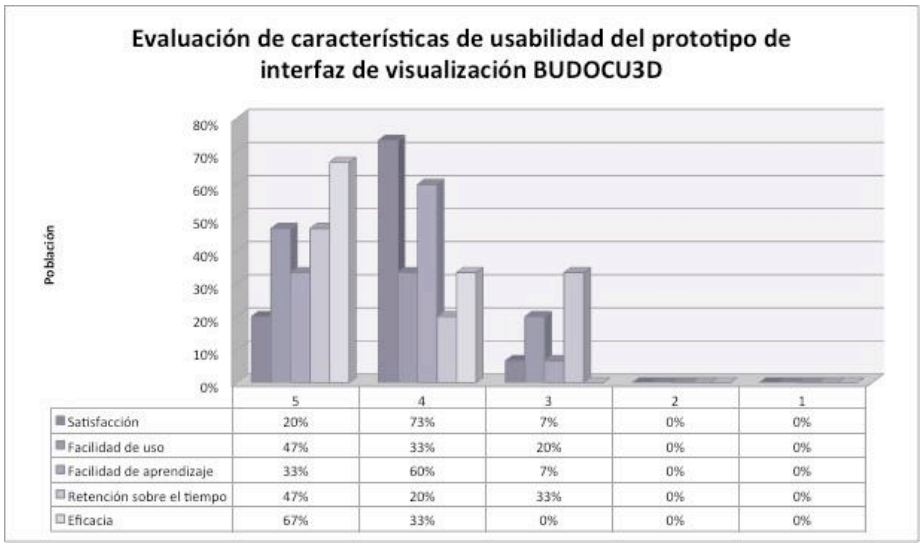
3.2.2 Resultados de pruebas de usabilidad al prototipo de interfaz

En esta sección se muestran los resultados experimentales después de haber aplicado cuestionarios de evaluación de usabilidad al prototipo de interfaz. En el Apéndice A de esta tesis es posible conocer de forma detallada el cuestionario aplicado.

Las pruebas fueron realizadas con una población de 15 usuarios, 6 del género femenino y 9 del género masculino, con un nivel de estudios profesionales y de posgrado, de ocupación en áreas administrativas y académicas, de una edad de entre 20 y 45 años. Estas personas interactúan diariamente con sistemas de información por computadora.

Se usó una escala *Likert* [25] de 5-1, donde 5 significa estar totalmente de acuerdo y 1 estar en total desacuerdo; se agruparon las preguntas de acuerdo a las

características de usabilidad y se obtiene el promedio para cada una de ellas. Los resultados se representan de acuerdo al porcentaje de la población encuestada. Ver Figura 15.



	5	4	3	2	1
Satisfacción	20%	73%	7%	0%	0%
Facilidad de uso	47%	33%	20%	0%	0%
Facilidad de aprendizaje	33%	60%	7%	0%	0%
Retención sobre el tiempo	47%	20%	33%	0%	0%
Eficacia	67%	33%	0%	0%	0%

Figura 15 Evaluación de características de usabilidad para el prototipo de interfaz de visualización BUDOCU3D

La Figura 15 es la gráfica de resultados de evaluación obtenida después de haber realizado los cuestionarios de usabilidad la cual se detalla a continuación:

Entre el 60% de la población califica con 5 a las características de usabilidad del prototipo de visualización, recordando la escala propuesta donde 5 es estar de totalmente de acuerdo y 1 en total desacuerdo.

3.3 Diagrama general de casos de uso

La Figura 16 muestra el diagrama general de casos de uso que representa las acciones que un usuario puede realizar en la interfaz, se describe la especificación de cada uno de los casos de uso para conocer cómo se ejecuta cada una de las tareas en el sistema. El prototipo sólo considera un usuario, es decir, no se implementan diferentes niveles de acceso.

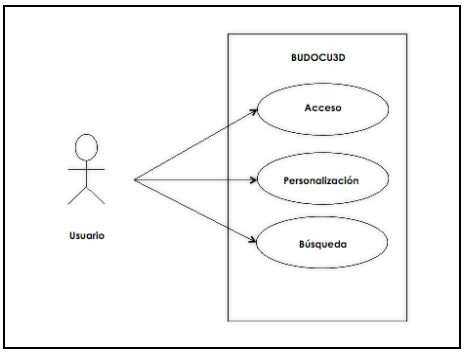


Figura 16 Diagrama general de casos de uso

En la **Tabla 2** se describen de forma general las acciones que el usuario puede realizar para el caso de uso **acceso,** así como el flujo de eventos de la acción.

Tabla 2. Descripción de eventos del caso de uso acceso.

Caso de uso	Descripción	Flujo de eventos
Acceso a cubo de colecciones	Tarea que le permite a un usuario tener acceso al cubo donde se encuentran representadas las colecciones a visualizar.	Despliegue de la pantalla del cubo de colecciones a representar, donde el usuario selecciona la colección a la que desea ingresar.
Acceso a cubo de clases	Tarea que le permite a un usuario tener acceso al cubo de clases de una colección.	Despliegue de la pantalla del cubo de clases de las colecciones a representar donde el usuario selecciona a qué clase desea ingresar.
Acceso a clases	Tarea que le permite a un usuario tener acceso a la pantalla de sub clases de una colección.	Despliegue de la pantalla de clases de una colección donde el usuario selecciona a qué sub clase desea ingresar
Acceso a visualización de documentos	Tarea que le permite a un usuario visualizar algún documento de una colección.	Despliegue de la pantalla de vista previa de un documento así como la liga de acceso al documento completo.

En la **Tabla 3** se describen de forma general las acciones que el usuario puede realizar para el caso de uso **personalizar** así como el flujo de eventos de la acción.

Tabla 3. Descripción de eventos del caso de uso personalizar.

Caso de uso	Descripción	Flujo de eventos
Personalización de color de cubo de colecciones	Tarea que le permite a un usuario personalizar el color de las caras del cubo de colecciones.	Despliegue del panel de personalización donde el usuario elige un color nuevo para las caras del cubo de colecciones.
Personalización de color de títulos de colecciones	Tarea que le permite a un usuario personalizar el color de los títulos correspondientes a una colección.	Despliegue del panel de personalización donde el usuario elige un color nuevo para el título de las colecciones.
Personalización de color de clase	Tarea que le permite a un usuario personalizar el color para cada una de las clases correspondientes a una colección.	Despliegue del panel de personalización donde el usuario elige un color nuevo para las clases de las colecciones.
Personalización de color de secuencia	Tarea que le permite a un usuario personalizar el color de la secuencia correspondiente a la navegación.	Despliegue del panel de personalización donde el usuario elige un color nuevo para la secuencia de navegación.
Personalización de color de documento	Tarea que le permite a un usuario personalizar el color del documento que quiere visualizar de una colección.	Despliegue del panel de personalización donde el usuario elige un color nuevo para los documentos de las colecciones.

En la **Tabla 4** se describen de forma general las acciones que el usuario puede realizar para el caso de uso **búsqueda,** así como el flujo de eventos de la acción.

Tabla 4. Descripción de eventos del caso de uso búsqueda.

Caso de uso	Descripción	Flujo de eventos
Búsqueda por palabra exacta	Tarea que le permite a un usuario realizar una búsqueda en base a una palabra exacta durante el proceso de navegación por las colecciones.	Despliegue del panel de búsqueda, donde el usuario introduce la palabra exacta que desea encontrar y dar clic sobre el botón **mostrar** para visualizar todas las coincidencias encontradas.
Búsqueda por múltiples palabras	Tarea que le permite a un usuario realizar una búsqueda utilizando palabras múltiples durante el proceso de navegación por las colecciones.	Despliegue del panel de búsqueda, donde el usuario introduce las palabras que desea encontrar y dar clic sobre el botón **mostrar** para visualizar todas las coincidencias encontradas.

3.4 Diagramas de clases de BUDOCU3D

El propósito de mostrar este diagrama de clases es el de representar los objetos fundamentales del sistema y su modelo de programación. La clase define el ámbito de definición de un conjunto de objetos donde cada objeto pertenece a una clase y es instanciado por la misma.

El modelo de programación consta de tres paquetes principales: eventos, utilerías y modelos. En el paquete eventos se encuentran alojadas clases que permiten manejar los eventos que el usuario puede realizar en la interfaz como por ejemplo, visualización de un documento, personalización de elementos de la interfaz o búsqueda de información.

El paquete “utilerías” aloja clases encargadas de manejar características del texto de la interfaz tales como colores, apariencias o tratamiento de documentos que alimentan a la interfaz de visualización.

Por último, pero no menos importante está el paquete de modelos, el cual aloja a las clases encargadas de la representación de los objetos de la interfaz en un ambiente 3D, se definen características de tamaño, movimiento y secuencia.

Las Figuras 17, 18 y 19 muestran los diagramas de clases por cada uno de los paquetes que conforman el sistema de visualización BUDOCU3D.

Figura 17 Diagrama de clases del paquete eventos BUDOCU3D

Figura 18 Diagrama de clases del paquete utilerías BUDOCU3D

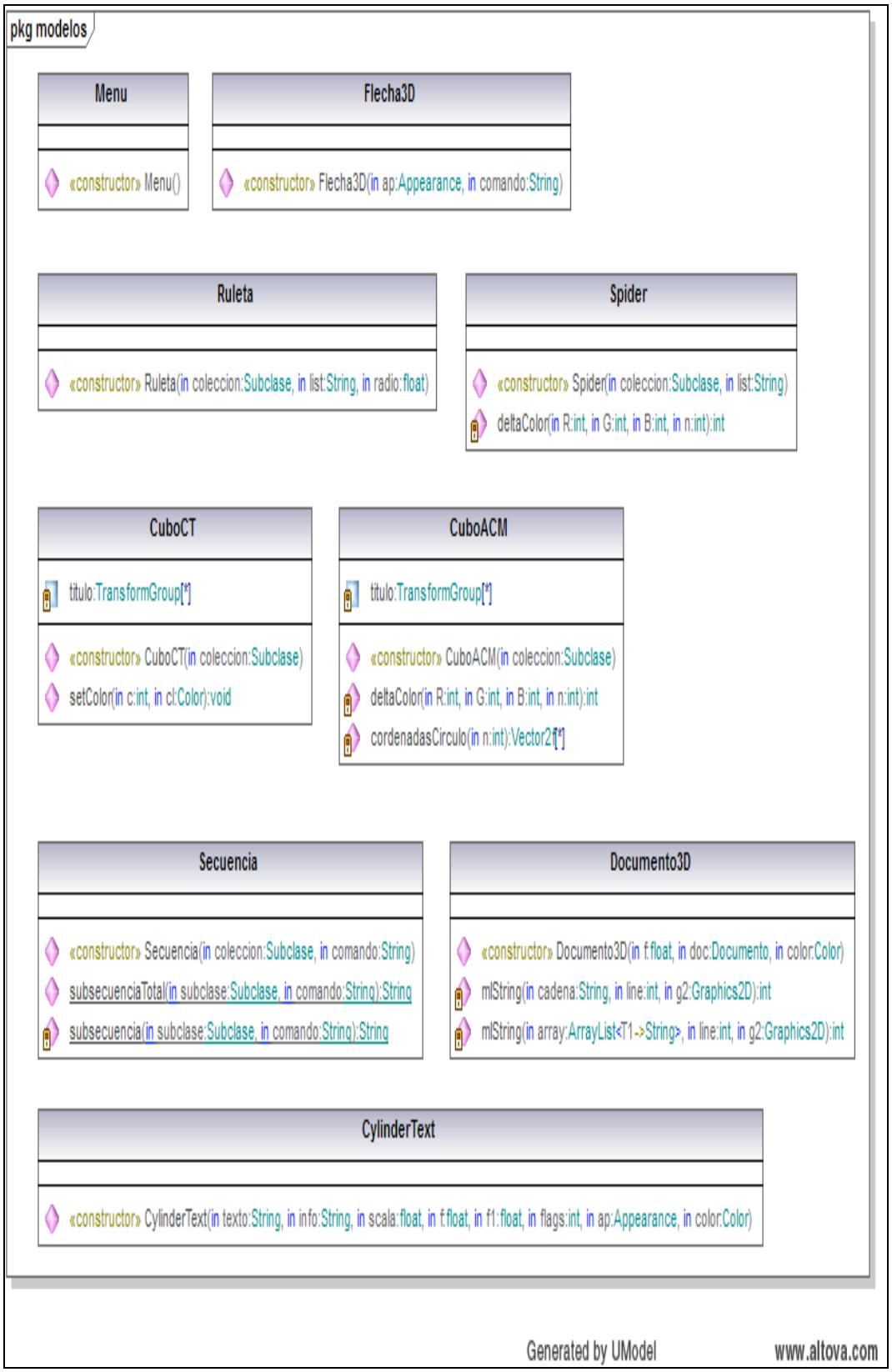


Figura 19 Diagrama de clases del paquete modelos BUDOCU3D

Capítulo 4

Implementación

4.1 Breve descripción de BUDOCU3D

BUDOCU3D se ha desarrollado como una interfaz de visualización de información que apoya la navegación por colecciones y permite localizar documentos de interés, así como recuperar información a través de la implementación de técnicas de búsqueda basadas en el empate de cadenas. Es una interfaz personalizable, es decir, el usuario puede seleccionar alguna combinación de colores definidos por omisión o modificar de acuerdo a sus preferencias algunos de los elementos de la interfaz.

4.1.1 Elementos de la interfaz de visualización

BUDOCU3D es una interfaz de visualización de información para documentos de tesis de maestría, reportes técnicos y registros OAI-PMH [21], aunque no es exclusivo ya que es posible representar cualquier tipo de documento que cumpla con el protocolo OAI-PMH y el estándar de metadatos Dublin Core [22]. La visualización de BUDOCU3D es representada jerárquicamente.

En las caras del primer cubo se localizan los títulos de las colecciones almacenadas, es ahí donde el usuario observa cilindros que representan las clases de la colección.

Al seleccionar alguna de las clases se despliegan las sub clases y los documentos contenidos así como una vista previa de los documentos que permite conocer algunos datos como el título, autores y descripción del documento además de poder ser direccionado hacia la ubicación original dentro de la colección.

A continuación se enlistan los elementos que componen la interfaz así como una breve descripción.

1. **Cubo de colecciones**

   Muestra en cada una de las caras del cubo, el título de las colecciones que es posible visualizar.

2. **Cubo de clases**

   Permite visualizar el cubo de clases donde se muestran las clases para cada colección.

3. **Clases**

Se muestran las clases que conforman a una colección.

4. **Subclases**

   Es posible visualizar las sub clases de una clase e ir adentrándose en la jerarquía hasta encontrar el documento de interés.

5. **Visualización**

   Esta parte de la interfaz muestra cómo es la vista previa de un documento de la colección, donde es posible conocer el título, autor y descripción.

6. **Enlace a URL**

Permite que, desde la vista previa del documento, sea posible salir y revisar el texto completo.

7. **Esquema de navegación**

Indica cual es la secuencia en la que se encuentra el usuario en la interfaz de visualización. Es la ruta desde la raíz hasta la clase o documento actual.

8. Búsqueda

Permite buscar palabras exactas o por múltiples palabras en la interfaz de visualización.

9. Personalización

Es posible para los usuarios personalizar los elementos de la interfaz de acuerdo a sus necesidades de visualización.

4.1.2 Implementación de la interfaz

En esta sección se habla acerca de los elementos que conforman la interfaz, la cual se forma por las siguientes secciones: visualización, personalización y búsqueda de documentos.

4.1.2.1 Datos de entrada para BUDOCU3D

Los registros a visualizar como ejemplo de utilización de la interfaz pertenecen a una colección que se encuentra formada por archivos bajo el estándar de Dublin Core (DC) [21] y OAI-PMH (Open Archives Initiative – Protocol for Metadata Harvesting) [22] que es un protocolo de comunicación utilizado para la transmisión de metadatos.

Los metadatos pueden ser definidos como datos sobre otros datos, es un término usado en la era de internet para la información que los bibliotecarios tradicionalmente colocaban en catálogos, se refiere a la información descriptiva sobre un recurso en la web.

Un registro de metadatos consiste en un conjunto de atributos o elementos necesarios para describir la fuente en cuestión, la relación entre un registro de metadatos y el recurso que describe puede darse de una de las siguientes formas:

1) Los elementos pueden estar en un registro separado del documento o 2) Los metadatos pueden estar incluidos en el propio recurso.

El estándar de metadatos (DC) [22] es un conjunto de elementos para describir una gama amplia de recursos de red. La adopción a gran escala de estándares y prácticas descriptivas para los recursos electrónicos mejora la recuperación de los recursos relevantes en cualquier contexto donde la recuperación es crítica.

La semántica de DC se ha establecido por un grupo internacional e interdisciplinario de profesionales de biblioteconomía, informática, codificación textual y otros campos teórico – prácticos relacionados.

Se utiliza como caso de estudio el sistema de clasificación ACM Computing [23], permite una organización temática de los documentos. En su versión 1998 se estructura jerárquicamente en cuatro niveles: tres niveles exteriores y un cuarto nivel no codificado de descriptores.

La interfaz de visualización está diseñada para interpretar un tipo de documento definido DTD, que incluye la descripción de la estructura y sintaxis en XML de la organización de la colección a visualizar.

El objetivo principal de utilizar un DTD es preservar una estructura en común y mantener la consistencia entre todos los documentos que utilizan la misma DTD.

De esta manera, dichos documentos pueden compartir descripción y forma de validación dentro de un grupo de colecciones de un tipo idéntico [24]. La Figura 20 muestra la estructura de un archivo de entrada para la interfaz de visualización

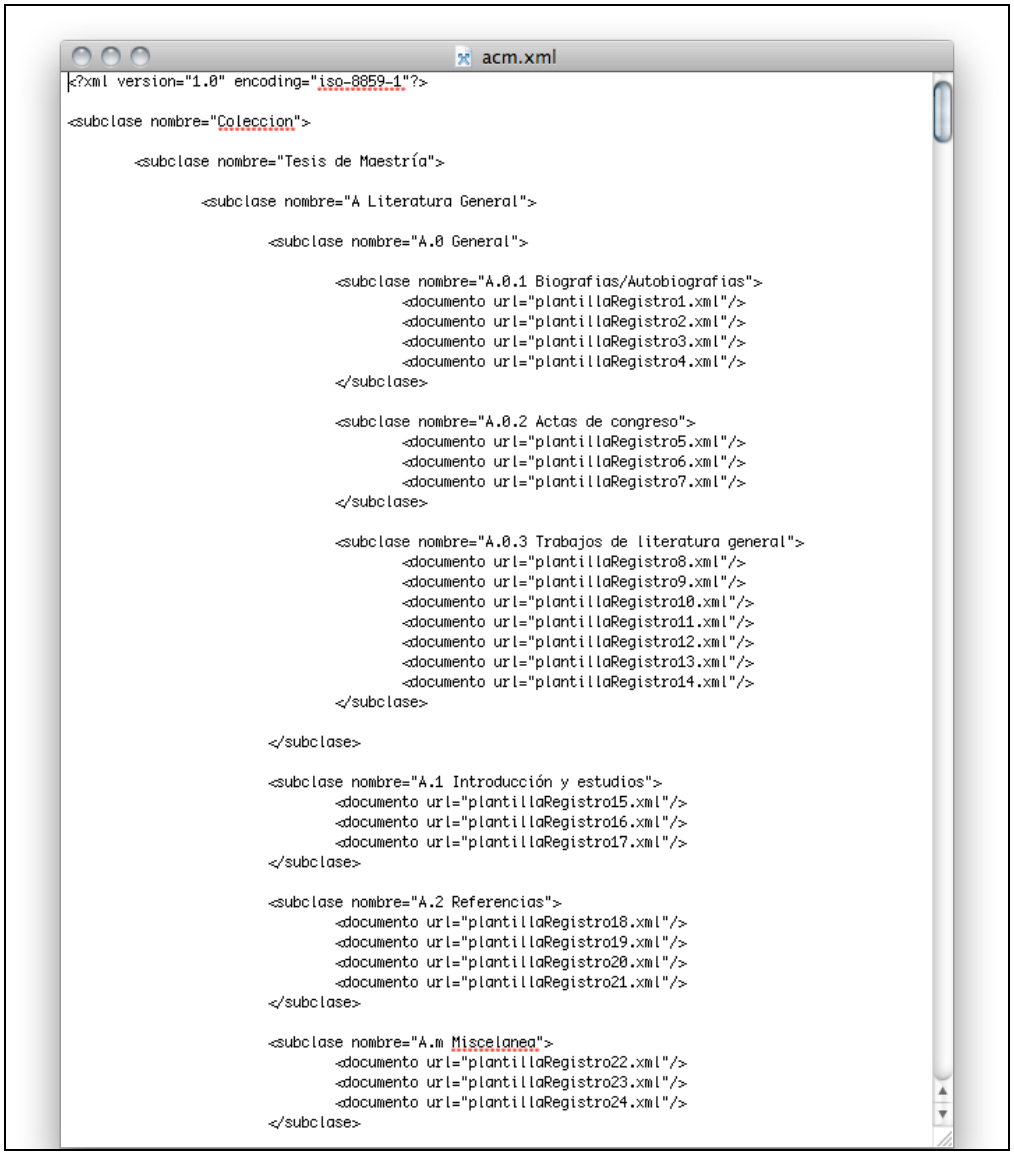


Figura 20 Representación XML que interpreta el sistema de visualización**.**

El archivo XML muestra un esquema de representación donde en base a clases y ***n*** número de subclases, es posible representar la jerarquía en la que se encuentra organizada una colección.

4.1.2.2 Visualización

El proceso de visualización comienza con el ingreso al cubo de colecciones donde se observa el título. Cuando un usuario da clic sobre un título, ingresará al cubo de clases el cual permite conocer las clases en las que se encuentra

clasificada la colección. Al elegir una de las clases de acuerdo a las sub clases que ésta contenga, es posible encontrar documentos con temas específicos y obtener como resultado final de este procedimiento la visualización de un documento en particular y sus datos.

De la Figura 21 a la Figura 24 se representa una secuencia de las imágenes que vería un usuario al interactuar con la interfaz.

Figura 21 Interfaz de cubo de colecciones

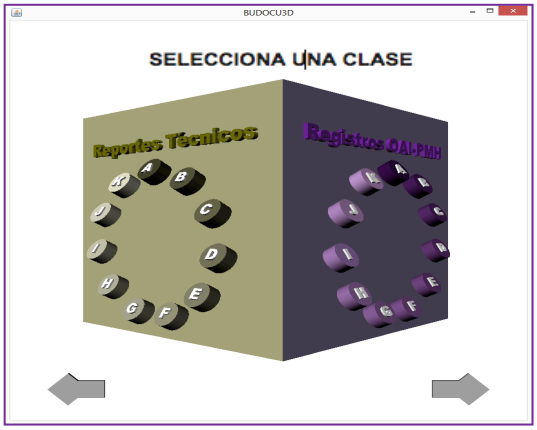


Figura 22 Interfaz de cubo de clasificaciones

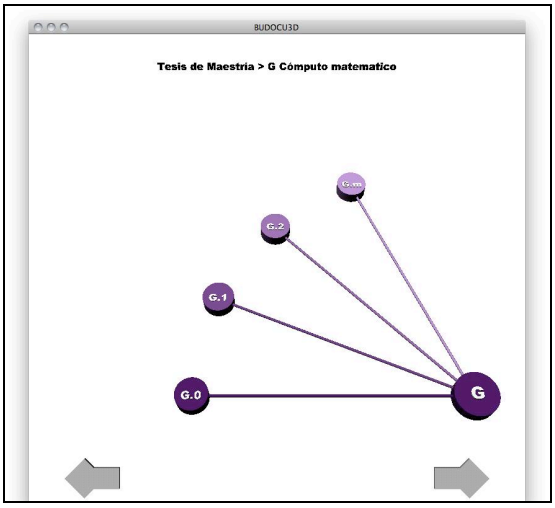


Figura 23 Interfaz para clases de colecciones

Figura 24 Interfaz para visualización de documentos

4.1.2.3 Personalización de interfaz

El panel de personalización está diseñado para que el usuario sea capaz de cambiar los colores correspondientes a cada uno de los elementos de la interfaz o seleccionar entre alguna de las combinaciones predefinidas.

Es posible cambiar el color de las caras de los cubos, de las clases de una colección, el color de letra de una colección y de una clase, así como el color y el fondo de la visualización del documento. En la Figura 25 se muestra el panel de personalización de la interfaz y en la Figura 26 se muestra la ventana de la

interfaz antes y después de haber realizado el cambio de colores.

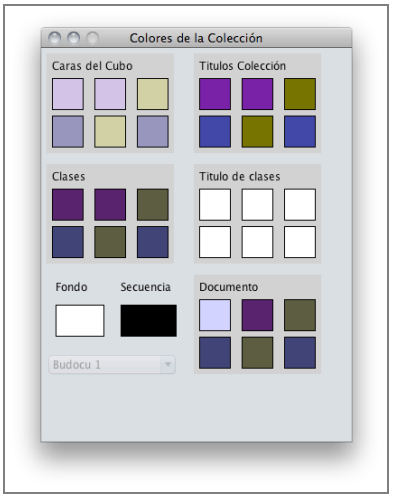


Fig. 25 Panel de personalización de la interfaz de visualización

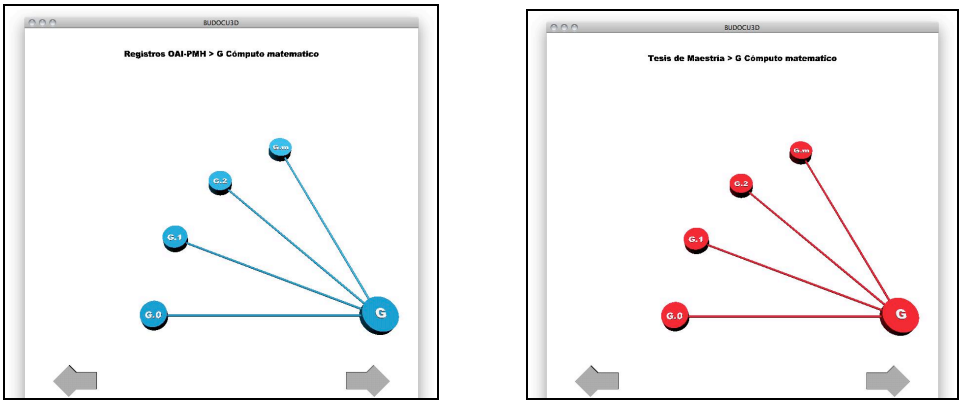


Fig. 26 Ventana de sub clases de colecciones después de realizar cambio de color

4.1.2.4 Panel de búsqueda en interfaz

El panel de búsqueda, el cual se muestra en la Figura 27 está diseñado para que un usuario pueda ingresar una palabra exacta o un grupo de palabras y se muestre una lista que cuando el usuario de clic sobre el término encontrado, sea re-direccionado a la parte de la interfaz donde se encuentran las ocurrencias del término en la secuencia de navegación. En la Figura 27 se muestra una ubicación de la interfaz encontrada por el panel de búsqueda.

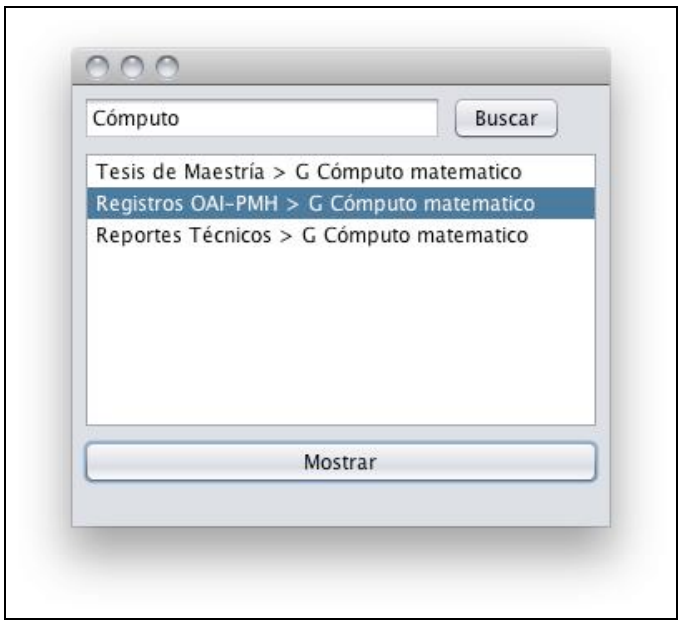


Figura 27 Panel de búsqueda en la interfaz de visualización

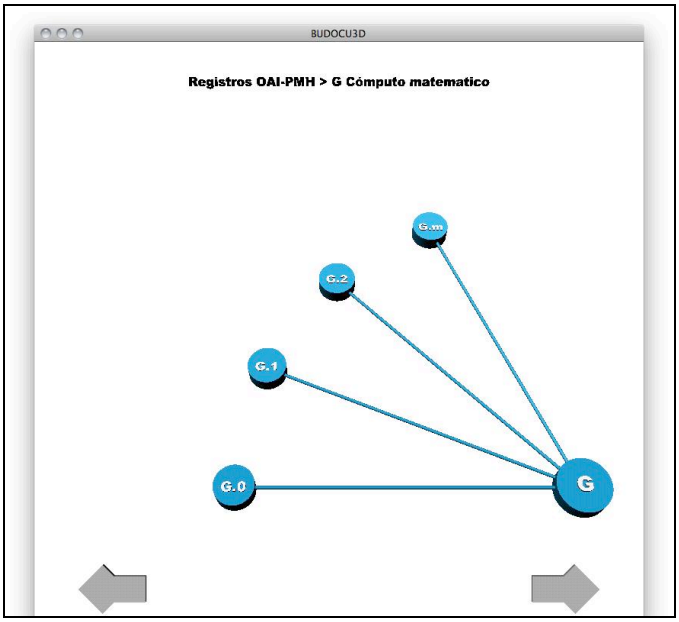


Figura 27 Panel de búsqueda en la interfaz de visualización

4.2 Procesamiento de los documentos

Cuando una colección no se encuentre clasificada, se deberá seguir la siguiente serie de pasos para procesar los documentos: 1) generar una representación XML

que sirva de entrada a la interfaz; 2) realizar la visualización de la colección, haciendo uso de una herramienta de utilidades creada de forma independiente de la interfaz; 3) generar un archivo en lenguaje XML capaz de ser interpretado por la interfaz de visualización. Estos pasos son esquematizados en la Figura 28.

4.2.1 Generación de vector de palabras clave

El análisis de los documentos comienza con la depuración de los datos para obtener las palabras claves, a continuación se enuncian los pasos a seguir para realizar esta tarea en la herramienta de procesamiento que se muestra en la Figura 28.

Paso 1:

Agregar documentos a procesar: seleccionar el botón "Agregar documentos" y una vez que han sido seleccionados dar clic sobre el botón "Cargar documentos" para procesarlos.

Paso 2:

Depurar documento: seleccionar la opción de "Depurar documentos" donde es posible seleccionar cuáles son los metadatos que se desean extraer, así como eliminar palabras vacías (artículos, pronombres, preposiciones, puntuación, números y acentos) y finalmente generalizar el texto en mayúsculas, minúsculas o mantener el texto original.

Paso 3:

Crear vector de palabras clave: seleccionar la opción de crear lista de palabras para generar el vector de palabras claves, el cual es necesario para generar el archivo con extensión arff que sirve como entrada al proceso de agrupamiento.

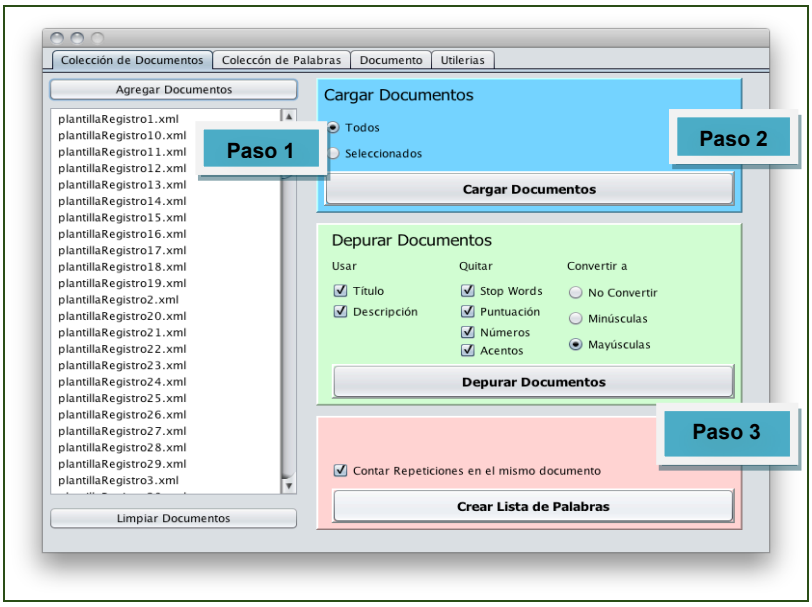


Figura 28 Herramienta de procesamiento de documentos.

4.2.2 Caracterización de los documentos

La extracción de características permite obtener el modelo vectorial para representar al documento con base en el título, autor y descripción.

El modelo vectorial se ha convertido en una herramienta estándar en sistemas de recuperación de información, se basa en una idea relativamente simple: dado un grupo de términos de un documento, no todos ellos son igualmente importantes para describir los contenidos, esto conduce a la asignación de pesos numéricos mayores o iguales a cero que se asigna a cada término o palabra de un documento De esta forma, el documento puede ser representado por el vector

$W_{ij} = (W_{1j}, W_{2j}, \ldots, W_{nj})$ [25].

Si N es el número de documentos de la colección y t el número de términos o palabras presentes; el conjunto de documentos puede ser representando por una matriz $W=Wij$ de dimensión $t \times N$, donde cada columna representa un documento y cada entrada representa el peso de una palabra en un documento. Esta matriz es conocida como *"matriz de términos – documentos"*.

Para el cálculo de los pesos o coordenadas de los vectores se ha utilizado un *esquema TF-IDF*. El peso de un término en un documento se obtiene como producto de dos factores; el primero de ellos, conocido como *factor TF*, mide la frecuencia de aparición del término en el documento, mientras que el *factor IDF*, conocido usualmente como ***frecuencia inversa*** del documento, permite rebajar significativamente el valor de los pesos correspondientes a términos con poco valor discriminante por aparecer en muchos documentos de la colección [25].

4.2.3 Creación de archivo arff con características de documentos

La construcción del vector de características se realiza en la pestaña de "Colección de palabras" donde es necesario indicar si se desea construir el vector con todas las palabras claves o sólo con las de mayor frecuencia.

A continuación se enuncian los pasos a seguir:

Paso 1:

Selección de palabras clave: Seleccionar si se desean incluir todas las palabras o seleccionar un rango con un porcentaje de frecuencias significativo.

Paso 2:

Creación de arff: después de haber seleccionado las palabras a utilizar, se crea el archivo arff que contiene la información del vector de características, es decir, las palabras claves y su frecuencia.

Tras haber realizado los pasos de la sección 4.2.3, el archivo arff está listo para ser procesado por Weka y realizar el agrupamiento como se muestra en la Figura 29.

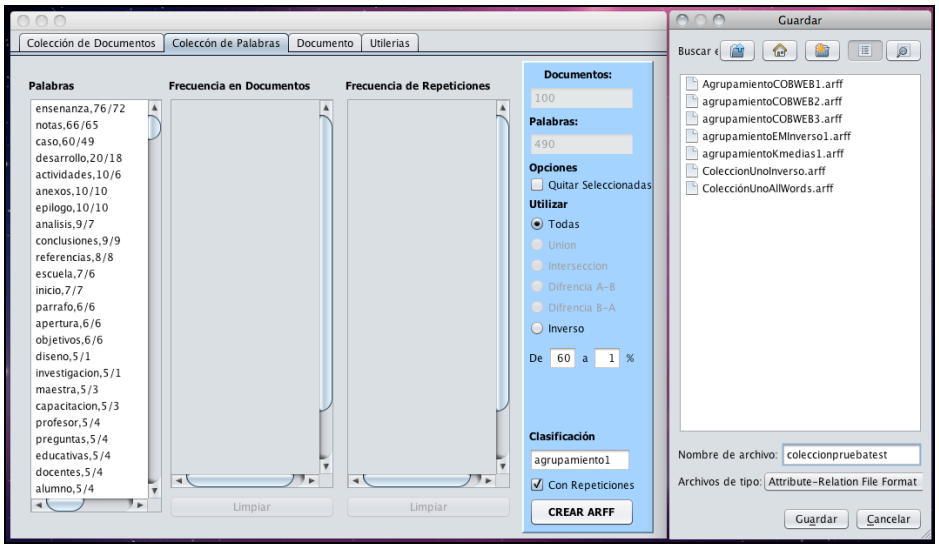


Figura 29 Herramienta de generación de archivo arff

4.3 Procesamiento de arff en Weka

Una vez que se ha obtenido el archivo arff que contiene las palabras claves encontradas en los documentos de la colección y su frecuencia de repetición, se puede procesar con Weka. Se realizaron pruebas con el algoritmo COBWEB [9] y K-Medias [9] para obtener grupos de documentos. A continuación se enuncian los pasos del procesamiento de un archivo arff en Weka que corresponde al procesamiento de 150 documentos extraídos del sitio de ReMeRi y otros provenientes del repositorio de la UPPuebla. La selección del algoritmo COBWEB se hizo con base en que no se necesita ingresar un número de grupos y es un algoritmo que proporciona un agrupamiento jerárquico, así también, se realizó la elección del algoritmo de K-Medias por ser uno de los algoritmos más utilizados para agrupar documentos. En esta tesis, es de interés evidenciar las diferencias encontradas entre ambos algoritmos con vistas a identificar su aplicación en contextos diversos.

Paso 1: Cargar archivo arff en el Explorer de Weka

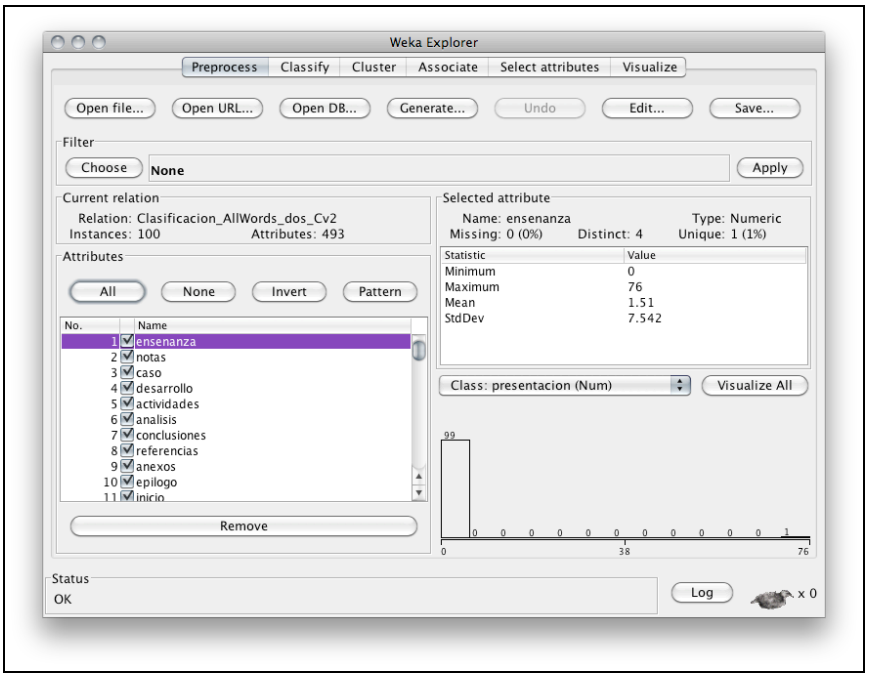


Figura 30 Procesamiento de archivo arff en Weka

Paso 2: Definir la configuración del archivo antes de aplicar algoritmo de agrupamiento.

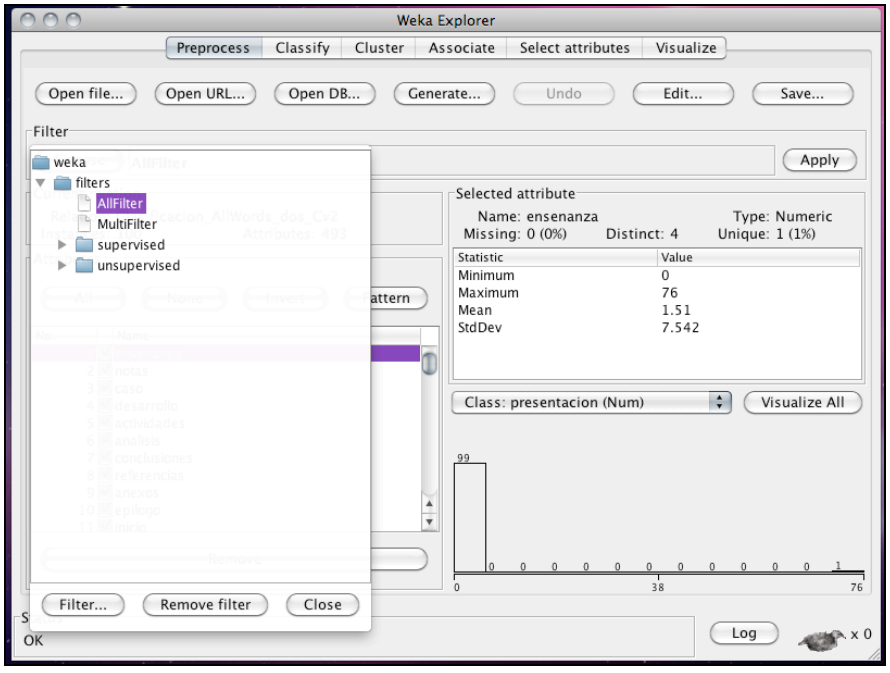


Figura 31 Configuración de archivo arff en Weka

Paso 3: Seleccionar algoritmo de agrupamiento para los datos del archivo arff

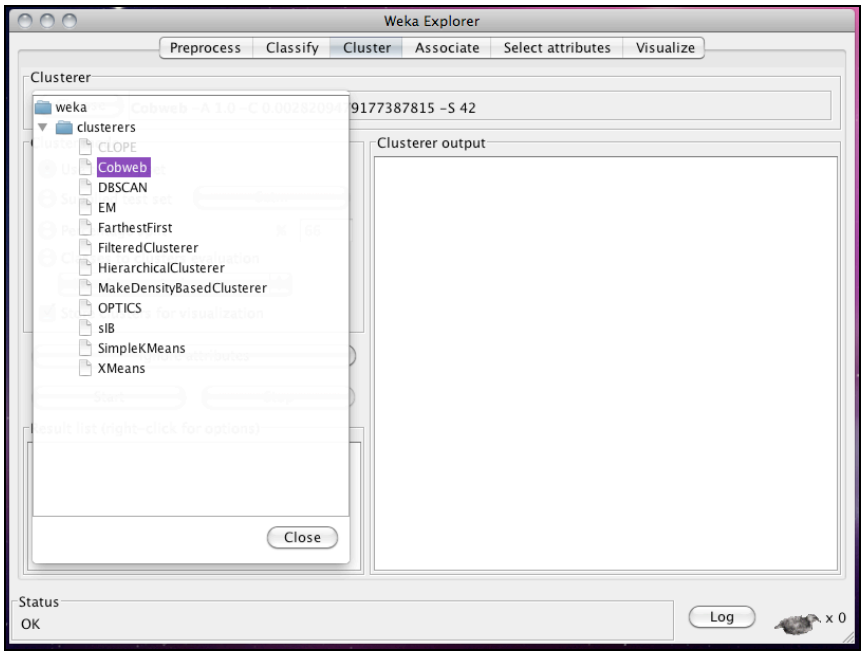


Figura 32 Selección de algoritmo de agrupamiento en Weka

Paso 4: Visualizar resultados de agrupamiento

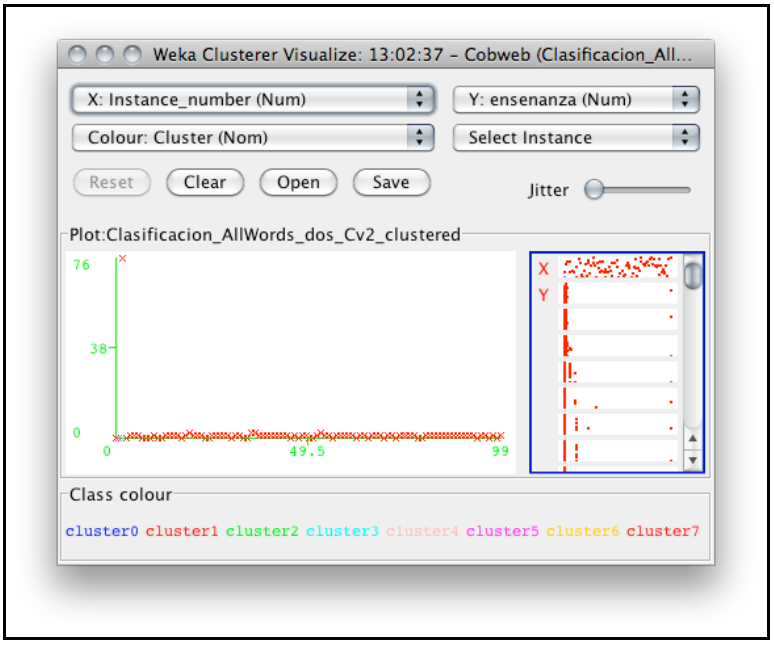


Figura 33 Visualización de resultados agrupamiento en Weka

Paso 5: Almacenar resultados de agrupamiento.

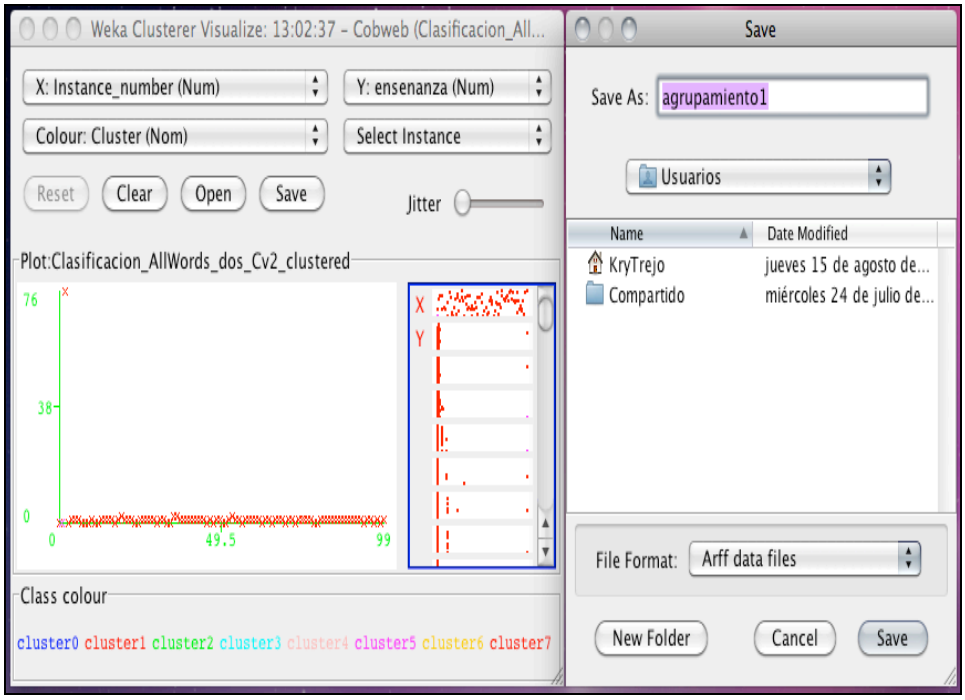


Figura 34 Resultados de agrupamiento en Weka

4.4 Interpretación de resultados del agrupamiento

Los resultados del agrupamiento fueron almacenados en un archivo arff generado por Weka que será procesado con la herramienta de utilidades de procesamiento de documentos. A continuación se enuncian los pasos a seguir para obtener un XML que sea posible representar en la interfaz de visualización.

Paso 1:

Separación de grupos: cargar el archivo arff de origen y el archivo arff de resultados para realizar la separación en grupos según el agrupamiento realizado por Weka dando clic en el botón **"Aplicar clúster"**. Cada carpeta en la que es particionada la información, representa un grupo que fue obtenido en el proceso de agrupamiento y aloja a los registros asociados a ese grupo, de tal manera que es posible representarlo de forma jerárquica para mostrar la información en la interfaz de visualización tal como se muestra en la Figura 35.

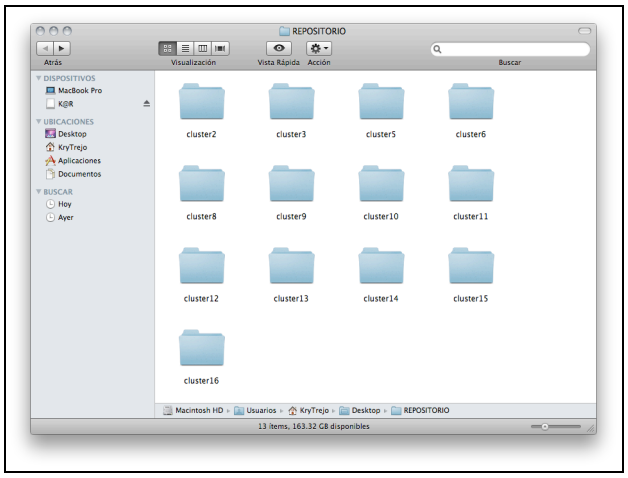


Figura 35 Visualización de la separación de grupos

Paso 2: Creación de archivo de representación en lenguaje XML: En el menú interfaz, se da clic sobre el botón crear archivo XML y se selecciona la carpeta donde se creó la separación de grupos para generar el archivo de representación XML necesario para la interfaz de visualización, como se muestra en la Figura 36 y Figura 37

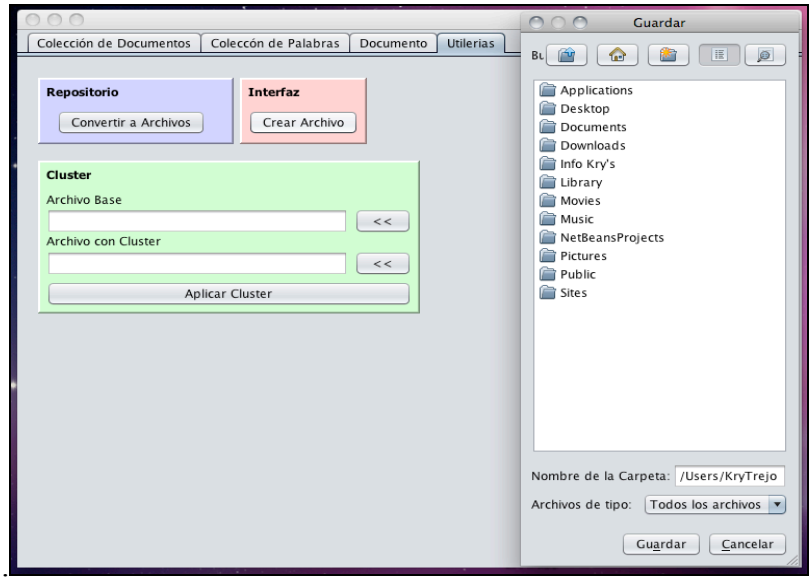


Figura 36 Creación de archivo XML

Figura 37 Generación del archivo de representación en XML

Capítulo 5

Pruebas

5.1 Prueba de usabilidad BUDOCU3D

En esta sección se muestran los resultados obtenidos después de haber aplicado un cuestionario de usabilidad a tres grupos de cinco personas, la población total de la prueba está compuesta por 6 personas de género femenino y 9 de género masculino con un nivel de estudios profesionales y posgrado, de ocupación en áreas administrativas y académicas, de una edad de entre veinte y cuarenta y cinco años que interactúan diariamente con sistemas de información por computadora.

Se propone como en las pruebas del prototipo de interfaz de visualización, una escala de 5-1 donde 5 significa estar de acuerdo y 1 estar en desacuerdo, se agrupan las preguntas de acuerdo a las características de usabilidad a la que correspondan y se obtiene el promedio para cada una de ellas y se representa de acuerdo al porcentaje de la población encuestada. La figuras 36 ilustra los gráficos con los resultados obtenidos.

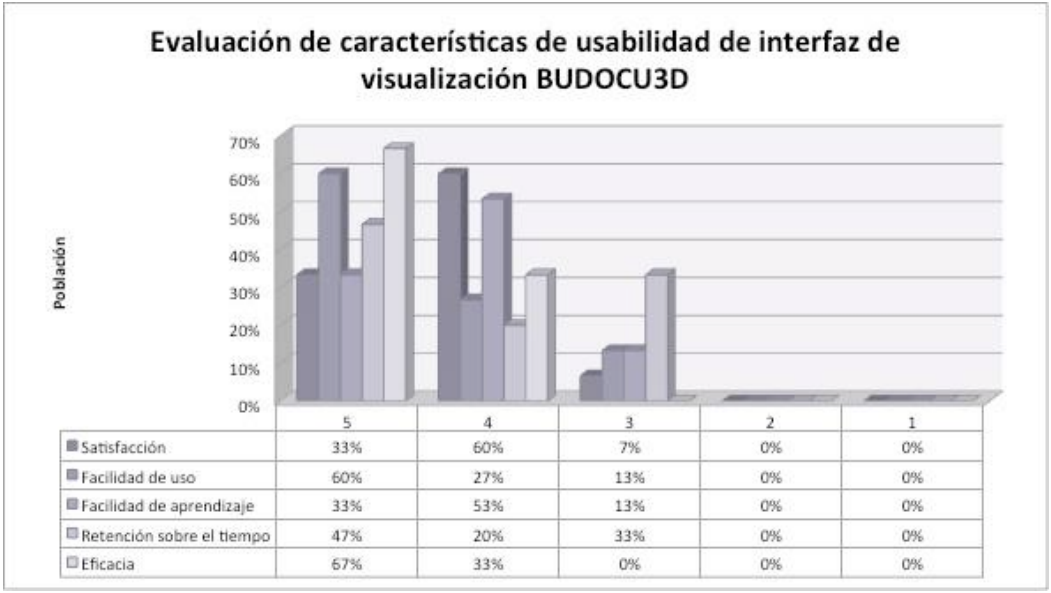


Figura 38 Gráfica de evaluación de características de usabilidad BUDOCU3D

La figura 38 es la gráfica de resultados de evaluación obtenida después de haber realizado los cuestionarios de usabilidad la cual se detalla a continuación:

Entre el 55% y 70% de la población califica con 5, 4 y 3 a las características de usabilidad de la interfaz de visualización, recordando la escala propuesta donde 5 es estar de acuerdo y 1 en total desacuerdo.

5.2 Pruebas de agrupamiento

Para estimar el mejor agrupamiento realizado por los algoritmos de agrupamiento COBWEB vs K-Medias, se realizó un previo agrupamiento manual el cual se ilustra en la tabla 5.

Grupo definido	Total de documentos
Ciencias sociales	69
General	17
Software	7
Negocios y economía	2
Metodologías	15
Organización de sistemas de computo	2
Aplicaciones web	3
Aplicaciones de computadora	6
Ciencias	4
Sistemas de información	17
Idiomas y literatura	8

Las pruebas se realizarán en el siguiente orden de pasos:

1. Preparación de los datos
2. Aplicación de los algoritmos COBWEB y K-Medias sobre el conjunto de datos en Weka
3. Análisis de resultados

5.3.2 COBWEB

El algoritmo COBWEB depende de dos parámetros que Weka propone por omisión 1) *acuity,* con un valor de 1.0 o utilidad de categoría, la cual mide el valor esperado de valores de atributos que pueden ser acertados a partir de la partición sobre los valores que se pueden acertar sin esa partición y 2) *cutoff que* se usa para parar la generación de nuevos nodos.

La tabla 6 muestra los resultados obtenidos después de realizar un grupo de pruebas a partir de la modificación del parámetro *cutoff* con un conjunto de estudio de 150 registros y 777 atributos, se calculó la similitud global existente entre los clusters generados de cada prueba para evaluar la calidad del agrupamiento, donde una menor distancia implica una mejor cohesión de los grupos.

Tabla 6. Agrupamiento con algoritmo COBWEB

Pruebas	Número de grupos
P1	11
P2	13
P3	13
P4	10
P5	11
P6	12
P7	12
P8	13
P9	11
P10	11
Promedio	**12**

La Tabla 6 muestra los resultados del agrupamiento realizado con el algoritmo COBWEB donde se observa que el número de grupos obtenidos se encuentra entre un rango de 10 y 13 con un promedio de 12 grupos para 10 pruebas realizadas.

5.3.2.1 K-Medias

Al algoritmo de K-Medias requiere como parámetro de entrada el número de grupos en el que debe segmentar los registros o Weka por omisión asigna 2. Por lo tanto, en las pruebas se ha seleccionado como número de grupos 11, el cual permite realizar una comparación con el agrupamiento realizado de forma manual mencionado anteriormente y calcular la métrica de F-*Measure* que es una métrica que combina *Precision* y *Recall,* lo cual evaluará la calidad del agrupamiento utilizando una colección de 150 registros y 777 atributos obteniendo los siguientes resultados:

Tablas 7. Agrupamiento algoritmo K-Medias Manual

PRUEBAS K-MEDIAS			
PRUEBAS	**PRECISION**	**RECALL**	**F-MEASURE**
P1	0.86	0.34	0.34
P2	0.86	0.33	0.34
P3	0.87	0.35	0.35
P4	0.87	0.36	0.36
P5	0.87	0.36	0.35
P6	0.86	0.34	0.34
P7	0.87	0.34	0.34
P8	0.87	0.37	0.36
P9	0.86	0.37	0.36
P10	0.87	0.33	0.35
PROMEDIO	0.87	0.35	0.35

La Tabla 7 muestra los resultados del agrupamiento realizado con algoritmo K-medias donde se muestran las métricas obtenidas de *Precision, Recall y F-Measure.*

Capítulo 6

Conclusiones

Se analizaron los algoritmos de agrupamiento para documentos COBWEB y K-medias para realizar agrupamiento cuando las colecciones a visualizar se encuentran desorganizadas, además se construyó una herramienta que permite obtener un esquema de representación de la información y su organización jerárquica en XML, el cual sirve de entrada para la interfaz de visualización propuesta.

Se diseñó un prototipo de interfaz de visualización donde se realizaron pruebas experimentales de usabilidad a un grupo de usuarios donde fue posible observar que el esquema de representación propuesto para la visualización de información es eficaz, el producto obtenido es la interfaz de visualización 3D.

La interfaz de visualización se implemento como un ambiente con herramientas y constructores de Java 3D que es flexible y personalizable, es decir, el usuario puede seleccionar combinaciones de color por omisión o puede personalizar su entorno de visualización, de tal manera que le permita desempeñar sus tareas de localización de información de una forma rápida y cómoda quedando como evidencia de satisfacción los resultados exploratorios de las pruebas de usabilidad.

Se implementaron técnicas básicas de búsqueda de datos dentro de la interfaz de visualización que permite al usuario localizar términos por palabra exacta o por múltiples palabras. Los resultados son mostrados en un panel que es lanzado al mismo tiempo que el sistema de visualización, permiten que el usuario seleccione de entre los resultados encontrados, a cuál quiere ser dirigido dando clic en el botón de mostrar.

Se concluye que se cumplieron satisfactoriamente el objetivo general y los objetivos específicos y se muestra una evidencia desarrollada para cada uno de ellos.

Como trabajo a futuro se propone extender la interfaz de visualización de manera que el usuario pueda seleccionar la posibilidad de migrarla a un ambiente de ejecución en web.

REFERENCIAS

[1] Proal C. Sistema uva: interfaces para visualización de grandes colecciones digitales. Tesis de maestría, Universidad de las Américas Puebla, Santa Catarina Mártir S/N, San Andrés Cholula, Puebla, México., Mayo 2002.

[2] Aguilar O. Enfocando la metáfora visual: Ópticas cognitivas I. Culturales, vol. VIII, núm. 16, Universidad Autónoma de Baja California, Mexicali, México., Julio-Diciembre 2012.

[3] Java SE Desktop Technologies Java API 3D. Consultada el 30 de septiembre de 2012 en:
http://www.oracle.com/technetwork/es/java/javase/overview/index.html

[4] NetBeans IDE. Consultada el 30 de septiembre de 2012 en:
https://netbeans.org/features/index.html

[5] Weka 3: Data Mining Software in Java. Consultada el 10 de enero de 2013 en:
http://www.cs.waikato.ac.nz/ml/weka/

[6] Vallejo S. Minería de datos, Universidad Nacional del Nordeste., Argentina 2006. Facultad de Ciencias Exactas, Naturales y Agrimensura. Trabajo de adscripción. Minería de datos.
http://exa.unne.edu.ar/depar/areas/informatica/SistemasOperativos/Mineria_Datos_Vallejos.pdf

[7] Hernández E. Algoritmo de clustering basado en entropía para descubrir grupos en atributos de tipo mixto. México, D.F., Agosto de 2006. CINVESTAV. Departamento de ingeniería eléctrica. Tesis de maestría.
http://www.cs.cinvestav.mx/TesisGraduados/2006/tesisEdnaHernandez.pdf

[8] Pascual D., Pla F., Sánchez S. Algoritmos de agrupamiento. Departamento de Computación, Universidad de Oriente, Santiago de Cuba, Cuba., 2008.

[9] Bedregal C. Agrupamiento de datos utilizando técnicas MAM-SOM. Universidad Católica de San Pablo, Perú., 2008.

[10] Terrádez M. Análisis de conglomerados. Ingeniería en informática. Universidad Oberta de Catalunya. Cataluña, España., 2013. Tesis de ingeniería. http://users.dcc.uchile.cl/~cbedrega/publications/Tesis.pdf

[11] Hoon M., Imoto S., Nolan J., Miyano S. Open Source Clustering Software. National Center for Biotechnology Information, US., 2004. Pag. 1454.

[12] Garre M., Cuadrado J., Rodríguez D. Comparación de diferentes algoritmos de clustering en la estimación de coste en el desarrollo de software. Departamento de ciencias de la computación. Ingeniería informática. Universidad de Alcalá, España., 2008.
http://ecd.es/IMG/pdf/GarreVol3Num1.pdf

[13] Rheingans P., Landreth C. Perceptual principles for effective visualizations. US EPA Visualization Center, MCNC, US., 2005.

[14] Castro S., Martig S. Visualización y computación gráfica. Instituto de Investigación en Ciencia y Tecnología Informática. Universidad Nacional del Sur, Buenos Aires, Argentina., 2007. Artículo científico.

[15] Bansal K., Sood S. Data Visualization A Tool of Data Mining. Himachal Pradesh University, Shimla, India., 2011.

[16] Bengochea L. Patricio M. Sistemas de visualización para bibliotecas digitales. Universidad de Alcalá, Madrid, España., 2005. Ciencias de la computación. Artículo científico.

[17] Red Mexicana de Repositorios Institucionales. Consultada el 15 de noviembre de 2011 en:
http://www.remeri.org.mx/remeri/premeri.html

[18] Sanchez J. Esquema de visualización de información para las publicaciones que se encuentran dentro de la colección ReMeRI. Universidad de las Américas Puebla, Santa Catarina Mártir S/N, San Andrés Cholula, Puebla, México., 2002.

[19] Shneiderman B. Designing the user interface. Strategies for effective human computer interaction. Addison – Wesley. 1998.

[20] Vega R., Rodriguez Z., Justo Y. Procedimiento para realizar pruebas de usabilidad. Universidad de las ciencias informáticas, Carretera a San Antonio de los Baños, km 2½, Torrens, Boyeros, La Habana, Cuba., 2010.

[21] Open Archives Initiative Protocol for Metadata Harvesting. Consultado en noviembre de 2011 en:
http://www.openarchives.org/pmh/

[22] The Dublin Core Metadata Initiative. Consultado en noviembre de 2011 en:
http://dublincore.org/

[23] ACM Computing Classification System. Consultado en noviembre de 2011 en:
http://www.acm.org/about/class/1998

[24] Cobo A., Rocha R. Desarrollo de una aplicación para la gestión, clasificación y agrupamiento de documentos económicos con algoritmos bio-inspirados. Departamento de Matemática Aplicada y Ciencias de la Computación, Universidad Cantabria. Av los Castros, 39005 Santander, Cantabria, España. 2010.

ANEXO A

CUESTIONARIO DE USABILIDAD

Prueba de Usabilidad BUDOCU 3D

Nivel máximo de estudios: ○ Primaria ○ Secundaria ○ Preparatoria ○ Profesional ○ Posgrado

Edad: ○ Menos de 20 ○ 20-25 ○ 25-30 ○ 35-40 ○ Más de 40

Sexo: ○ Masculino ○ Femenino

Ocupación: ○ Estudiante ○ Profesor ○ Administrativo ○ Directivo

Experiencia con sistemas de visualización de información: ○ Si ○ No

Fecha de la prueba: _____ (*dd/mm/aa*)

Instrucciones generales

Estimado usuario:

Se quiere probar la **usabilidad** de la interfaz de visualización BUDOCU3D. El objetivo de esta prueba es verificar si la interfaz propuesta cumple con sus tareas y si re resulta fácil de usar.

Le pedimos realice las tareas descritas en la página siguiente. Por favor lea cuidadosamente cada una antes de empezar. Después, se le pedirá conteste un cuestionario. Estamos muy interesados en sus impresiones subjetivas y espontáneas, es por eso que se le pide manifieste en voz alta sus opiniones durante la prueba.

Si tiene algún comentario por favor comuníquelo al facilitador. Si ya no hay más preguntas, entonces inicie la prueba.

Se agradece de antemano su atención y tiempo.

Lista de tareas:

1. Cambiar el color de la secuencia de navegación.

2. Localizar la categoría de “Hardware” haciendo uso del panel de búsqueda.

Preguntas para evaluar la usabilidad de BUDOCU3D 1: localizar documento de una colección.

1) ¿Cómo calificaría la interfaz de visualización BUDOCU3D?

Buena	☐	☐	☐	☐	☐	Mala
Clara	☐	☐	☐	☐	☐	Confusa
Sencilla	☐	☐	☐	☐	☐	Difícil

2) ¿Le fue fácil cumplir con las tareas que se le pidieron realizar?

De acuerdo	☐	☐	☐	☐	☐	En desacuerdo

3) ¿Considera que la aplicación es fácil de utilizar?

De acuerdo	☐	☐	☐	☐	☐	En desacuerdo

4) ¿Considera que la información se encuentra organizada?

De acuerdo	☐	☐	☐	☐	☐	En desacuerdo

5) ¿La propuesta de colores por omisión para la interfaz es de su agrado?

De acuerdo	☐	☐	☐	☐	☐	En desacuerdo

6) ¿Considera que el tiempo empleado en el que realizo las tareas solicitadas fue corto ?

De acuerdo ☐ ☐ ☐ ☐ ☐ En desacuerdo

7) ¿Considera que los iconos mostrados en la interfaz son fáciles de comprender?

De acuerdo ☐ ☐ ☐ ☐ ☐ En desacuerdo

8) ¿Relaciona la interfaz con alguna actividad cotidiana que realiza?

De acuerdo ☐ ☐ ☐ ☐ ☐ En desacuerdo

9) ¿Considera que el panel de personalización es fácil de comprender?

De acuerdo ☐ ☐ ☐ ☐ ☐ En desacuerdo

10) ¿Considera que el panel de búsqueda es fácil de usar?

De acuerdo ☐ ☐ ☐ ☐ ☐ En desacuerdo

11) ¿Considera que es capaz de volver a interactuar con la interfaz en una segunda prueba y cumplir con mayor facilidad las tareas que se le pidieron realizar?

De acuerdo ☐ ☐ ☐ ☐ ☐ En desacuerdo

12) ¿Qué tiempo le llevo realizar las tareas?

_____
_____

13) ¿Surgieron dudas de cómo utilizar la aplicación?

○ Si ○ No ¿Cuáles?

_____
_____

14) ¿Tuvo algún problema al utilizar la aplicación 1? ○ Si ○ No ¿Cuál?

_____

_____
_____

15) ¿Cómo se siente después de realizar la prueba?

16) ¿Qué le gusto más? ¿Qué no le gustó? ¿Tiene alguna propuesta para mejorarlo?

¡MUCHAS GRACIAS!