



Universidad de las Ciencias Informáticas
Facultad 1

Sistema de Recomendación para un Proveedor de Servicio OAI-PMH.



Trabajo de Diploma para optar por el título de Ingeniero en Ciencias Informáticas.

Autor: Alberto González Ojeda

Tutores: Ing. Maikel Manuel Fernández Fernández

Ing. Luis Domínguez Cruz

La Habana, junio 2014

Declaración de autoría

Declaro ser autor del presente trabajo de diploma y otorgo a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

Para que así conste firmamos las presentes a los ____ días del mes de _____ del año 2014.

Tutores:

Ing. Maikel Manuel Fernández Fernández

Ing. Luis Domínguez Cruz

Autor:

Alberto González Ojeda

Datos de contacto

- **Ing. Maikel Manuel Fernández Fernández**
Universidad de las Ciencias Informáticas, La Habana, Cuba.
Correo: maikelm@uci.cu
- **Ing. Luis Domínguez Cruz**
Universidad de las Ciencias Informáticas, La Habana, Cuba.
Correo: ldominguezc@uci.cu

Dedicatoria

Dedico este trabajo a mi madre, por ser la única amiga que verdaderamente en las buenas y malas a estado a mi lado todo el tiempo incansablemente, gracias por confiar en mí en todo momento, este premio es tuyo, a mi virgencita por siempre escucharme y ser testigo de mi realidad y aun así ofrecerme fe para seguir adelante, a toda mi familia en especial a mi abuela Ena y a mi abuelo Juan que le hubiera encantado compartir conmigo este momento, gracias por su educación, a mi hermana linda que siempre estuvo a mi lado apoyándome, a mi tío Pitivity, gracias por los concejos, a mi padre que en los últimos años he sentido su apoyo, a mis sobrinitas que las quiero mucho y siempre me preguntaban por la escuela, a mis amigos, Ernesto, Rolnic, Fran Luis, El Piry, Aroche, O'Neill, Arianna, gracias por el apoyo. A todos los profesores y a las personas que de una forma u otra ayudaron y contribuyeron en mi superación profesional. A todos mis compañeros que siempre estarán en mi corazón. A ustedes les dedico este trabajo.

Agradecimientos

Le agradezco a la revolución por darme la oportunidad de pertenecer a esta gran universidad, a mis tutores; Maiquel Manuel Fernández Fernández y Luis Domínguez Cruz, por ser tan pacientes y haber dedicado tanto tiempo a la realización de este trabajo. A Ivis Cañizares Rivera por su ayuda y atención a lo largo de mi carrera. A mis amigos, gracias por siempre estar presente en los momentos más difíciles de mi carrera en especial a Ernesto, El Piry, Yoandri, O'Neill, Jorge David, El Guille, Leo, Calero y Arianna, gracias por la paciencia y el apoyo, a mi novia Adri, gracias por todo tu amor y dedicación, al piquete del Basket, a mis compañeros de grupo en especial a los de 14 y 1503, gracias por su aceptación y su amistad; de forma general gracias a todos aquellos que de una u otra forma me dieron su apoyo y ayuda para realizar este sueño que tanto trabajo me ha costado.

Gracias por todo.

Resumen

La recomendación es un proceso que actualmente cobra una gran importancia debido al aumento significativo de la información en todos sus escenarios y formatos. El presente trabajo describe el desarrollo de un sistema de recomendación basado en contenidos. El intercambio de información académica y científica, específicamente aquella que se presenta bajo el modelo de acceso abierto, constituye el dominio donde se aplica la solución propuesta. El producto final constituye un módulo para el sistema de gestión de contenidos Drupal 7.23, debido a la necesidad de integrar el sistema de recomendación con un proveedor de servicios OAI-PMH, previamente desarrollado con ese gestor de contenidos. El propósito final es recomendar documentos al usuario según sus preferencias. Para esto se realizan comparaciones entre documentos basadas en la métrica del coseno como medida de similitud, además se gestionan las entidades usuario, documento y clasificador para complementar el proceso. En el trabajo se realiza un estudio previo de los sistemas de recomendación, especialmente aquellos basados en contenidos. Se hace uso de la metodología *OpenUP* para la descripción y modelado del proceso de desarrollo y se expone el proceso de implementación. Finalmente se realizan pruebas funcionales y de carga y estrés para comprobar la calidad del sistema.

Palabras clave: acceso abierto, *dublin core*, *drupal*, *OAI-PMH*, proveedor de servicios, repositorio, sistemas de información.

Índice

| | |
|---------------------------------------------------------------------------------------------------------------------------------------------------|----|
| INTRODUCCIÓN | 1 |
| CAPÍTULO 1: CARACTERIZACIÓN DEL ENTORNO CONCEPTUAL Y TECNOLÓGICO DEL SISTEMA DE RECOMENDACIÓN PARA UN PROVEEDOR DE SERVICIOS OAI-PMH | 5 |
| INTRODUCCIÓN | 5 |
| 1.1.1. Preferencias | 5 |
| 1.1.2. Recomendación | 5 |
| 1.1.3. Sistema de recomendación | 6 |
| 1.2. ESTRUCTURA DE LOS SISTEMAS DE RECOMENDACIÓN | 6 |
| 1.2.1. Métodos de generación de recomendaciones | 6 |
| 1.2.2. Clasificación de los sistemas de recomendación | 7 |
| 1.2.3. Sistema de Recomendación Basado en Contenido | 8 |
| 1.2.4. Perfil de usuario | 9 |
| 1.3. PROTOCOLO Y ESTÁNDAR DE METADATOS | 9 |
| 1.4. MODELO VECTORIAL | 10 |
| 1.5. MEDIDAS DE SIMILITUD ENTRE DOCUMENTOS | 11 |
| 1.6. ANÁLISIS DE SOLUCIONES EXISTENTES | 11 |
| 1.6.1. Referral Web | 11 |
| 1.6.2. Fab | 12 |
| 1.6.3. Diva | 12 |
| 1.6.4. SlideShare | 12 |
| 1.6.5. ResearchGate | 12 |
| 1.6.6. Sistemas de recomendación desarrollados en la UCI | 13 |
| 1.7. APORTES DEL ESTUDIO DE LOS SISTEMAS DE RECOMENDACIÓN A LA PROPUESTA DE SOLUCIÓN | 13 |
| 1.8. METODOLOGÍA DE DESARROLLO DE SOFTWARE | 13 |
| 1.9. HERRAMIENTAS CASE | 14 |
| 1.9.1. Visual Paradigm | 15 |
| 1.10. UML LENGUAJE UNIFICADO DE MODELADO | 15 |
| 1.11. LA PILA TECNOLÓGICA DE DIMA | 15 |
| 1.11.1. MySQL | 15 |
| 1.11.2. Servidor Web Apache 2.2.21 | 16 |
| 1.11.3. Lenguaje de programación PHP 5.3.8 | 16 |
| 1.11.4. HTML | 17 |
| 1.11.5. CSS 3 | 17 |
| 1.11.6. Drupal | 18 |
| 1.12. ENTORNO DE DESARROLLO INTEGRADO (IDE) | 18 |
| 1.12.1. NetBeans | 18 |
| CAPÍTULO 2: PROPUESTA DE SOLUCIÓN DEL SISTEMA DE RECOMENDACIÓN PARA UN PROVEEDOR DE SERVICIO OAI-PMH. | 20 |
| 2.1. OBJETO DE AUTOMATIZACIÓN | 20 |
| 2.2. PROPUESTA DEL SISTEMA | 20 |
| 2.3. MODELO DE DOMINIO | 21 |

| | | |
|----------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------|-----------|
| 2.3.1. | Descripción de las clases del modelo de dominio..... | 21 |
| 2.4. | REQUISITOS..... | 22 |
| 2.4.1. | Requisitos Funcionales | 22 |
| 2.5. | REQUISITOS NO FUNCIONALES..... | 23 |
| 2.5.1. | Usabilidad..... | 23 |
| 2.5.2. | Disponibilidad | 23 |
| 2.5.3. | Confiabilidad..... | 23 |
| 2.5.4. | Fiabilidad | 23 |
| 2.5.5. | Legales..... | 24 |
| 2.5.6. | Soporte..... | 24 |
| 2.5.7. | Hardware..... | 24 |
| 2.6. | MODELO DE CASOS DE USO..... | 25 |
| 2.6.1. | Diagrama de casos de uso..... | 25 |
| 2.6.2. | Descripción de los casos de uso..... | 26 |
| 2.7. | ESTILOS ARQUITECTÓNICOS..... | 31 |
| 2.7.1. | Patrones arquitectónicos..... | 31 |
| 2.8. | PATRÓN DE DISEÑO | 33 |
| 2.8.1. | Patrones GOF | 33 |
| 2.8.2. | Patrones GRASP | 34 |
| 2.9. | MODELO DE DISEÑO..... | 35 |
| 2.9.1. | Diagramas de clases del diseño | 35 |
| 2.9.2. | Descripción de las clases del diagrama de clases de diseño | 36 |
| CAPÍTULO 3: IMPLEMENTACIÓN Y VALIDACIÓN DE LOS RESULTADOS DEL SISTEMA DE RECOMENDACIÓN PARA UN PROVEEDOR DE SERVICIOS OIA-PMH | | 37 |
| INTRODUCCIÓN..... | | 37 |
| 3.1. | MODELO DE IMPLEMENTACIÓN..... | 37 |
| 3.1.1. | Diagrama de componentes | 37 |
| 3.1.2. | DESCRIPCIÓN DE LOS COMPONENTES DEL DIAGRAMA | 38 |
| 3.2. | MODELO DE DESPLIEGUE..... | 39 |
| 3.3. | FUNCIONES IMPORTANTES | 40 |
| 3.4. | PRUEBAS DEL SISTEMA | 45 |
| 3.4.1. | PRUEBAS FUNCIONALES | 47 |
| 3.4.2. | RESULTADO DE LAS PRUEBAS | 51 |
| 3.5. | RESULTADOS DE LAS PRUEBAS DE CARGA Y ESTRÉS | 52 |
| 3.6. | PRUEBAS DE SEGURIDAD..... | 54 |
| 3.6.1. | Resultados de las pruebas | 54 |
| CONCLUSIONES GENERALES..... | | 56 |
| RECOMENDACIONES..... | | 57 |
| BIBLIOGRAFÍA Y REFERENCIAS BIBLIOGRÁFICAS | | 58 |
| GLOSARIO DE TÉRMINO | | 62 |
| ANEXOS..... | | 63 |

Introducción

Con las nuevas tecnologías y especialmente con el rápido crecimiento de Internet, los usuarios encuentran una gran variedad de contenidos de hipertexto y multimedia en los diferentes sitios *Web* existentes en Internet. Muchas veces, estos usuarios al no tener un conocimiento exacto de lo que buscan se sienten abrumados por el exceso de información que existe, por esta razón se ha hecho necesario algún tipo de ayuda para identificar aquellos archivos que les puedan ser útiles. Gracias al avance de las tecnologías de la información y las comunicaciones (TIC) y la inclusión de éstas en cada esfera de la vida, han sido desarrollados sistemas informáticos capaces de hacer recomendaciones de contenido a usuarios seguidores de determinados temas facilitando entre otras cosas la gestión del conocimiento.

En el caso de la información académica y científica, existe una situación similar, un crecimiento acelerado en el aspecto cuantitativo y por tanto en la dificultad por parte de las personas para mantenerse al día con la búsqueda. En este dominio específico se enmarca el movimiento de acceso abierto, que busca facilitar la distribución de la información científica sin barreras, tratando de cambiar la situación que imponen los grandes consorcios de investigación científica donde cada día aumentan los precios de los recursos de información e introducen barreras legales al conocimiento.

El movimiento de acceso abierto se ha encargado a través de los años de la socialización del conocimiento y del desarrollo de estándares abiertos para el intercambio de información independientemente de la diversidad tecnológica. Muchos han sido los servicios que se han implementado mundialmente utilizando *Dublin Core* (DC) como estándar de metadatos y *OAI-PMH* (*Open Access Initiative – Protocol for Metadata Harvesting*) como protocolo de comunicación, ambas ideas materializadas y potencializadas por este movimiento.

En la Universidad de las Ciencias Informáticas (UCI), en años anteriores (2010 - 2012), se desarrolló lo que inicialmente recibió el nombre de Portal de la Biblioteca Alma Mater (ABREU, 2011) y seguidamente pasó a llamarse biblioteca digital Arístides Rojas, sistema implementado gracias a los convenios de colaboración Cuba – Venezuela, utilizando como precepto el acceso abierto al conocimiento y las tecnologías de este tipo de movimiento. Luego se creó una versión denominada portal de la biblioteca de la UCI (BELLO, 2012), lo que fue evolucionando hacia un proveedor de servicios OAI-PMH, el cual se encuentra actualmente instalado en el Ministerio de Educación Superior (MES) bajo el nombre de DIMA (Directorio Integrado de Materiales en Acceso Abierto) (FERNÁNDEZ, 2014).

DIMA es un proveedor de servicios que contiene un cúmulo amplio de información que crece de forma acelerada, es muy complicado para el usuario mantenerse al tanto de las novedades de su interés sin una

funcionalidad que le sugiera los recursos específicos que pueden ser de su utilidad. La no existencia de un mecanismo de recomendación incide en la valoración que tiene el usuario del sistema y le dificulta la tarea de gestión de información y de actualización.

Por esta razón es que surge la idea de realizar un sistema de recomendación para un proveedor de servicio OAI-PMH cuya meta es lograr una aplicación capaz de ofrecer sugerencias de documentos a un usuario, obtenida a partir de sus preferencias.

Para generar las recomendaciones hay tres tipos fundamentales de sistemas que se adaptan al escenario donde se desempeña DIMA. El primero es basado en contenido donde el sistema recomienda objetos similares a otros que han sido del agrado del usuario en el pasado, el segundo es llamado algoritmo colaborativo, donde el sistema debe de recomendar a un determinado usuario objetos que han sido del agrado de otros usuarios con gustos similares a los suyos, y el tercero son los sistemas de recomendación basados en conocimiento, los cuales utilizan el conocimiento que proporciona el usuario sobre sus necesidades y que tiene el sistema sobre los productos para realizar las recomendaciones que mejor cubren las necesidades de los usuarios (BARRANCO, 2006).

Una premisa importante que rige la presente investigación es que el desarrollo del sistema de recomendación debe integrarse a una aplicación basada en la arquitectura del sistema de gestión de contenidos *Drupal*, en su versión 7.x y que emplea OAI-PMH para el intercambio de la información.

La situación que se presenta es la no existencia de un sistema de recomendación dentro de la aplicación DIMA. Esto trae consigo varias incidencias negativas. Es complicado para el usuario mantenerse actualizado de las novedades en los temas de su interés debido a que depende totalmente de la recuperación de información; esto implica que sea necesaria una alta precisión en la forma de expresar su necesidad de información. Por otra parte utilizar la recuperación de información por navegación es complicado debido al cúmulo de documentos existente. También se puede señalar el inconveniente para el usuario de tener que, sistemáticamente, realizar el proceso de búsqueda, probablemente para una misma consulta.

Basado en lo antes analizado, se llega a la conclusión de que la actividad de recomendación es sustantiva en cualquier sistema que gestione información, por esta razón y a partir de las dificultades ya mencionadas se origina el siguiente **problema de investigación**: ¿Cómo recomendar a los usuarios de un proveedor de servicios OAI-PMH los contenidos de su interés?

Para enfrentar el problema anterior se define como **objeto de estudio** los sistemas de recomendación, enmarcando el **campo de acción** en los sistemas de recomendación basados en contenido.

El **objetivo general** del presente trabajo es implementar un sistema de recomendación, que mediante el análisis de similitud entre documentos, permita sugerir contenidos a los usuarios de un proveedor de

servicios OAI-PMH.

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

- ❖ Construir el marco teórico conceptual de la investigación que permita la comprensión del escenario y la identificación de las técnicas a utilizar.
- ❖ Identificar las funcionalidades del sistema de recomendación.
- ❖ Describir las funcionalidades asociadas al sistema de recomendación.
- ❖ Implementar las funcionalidades identificadas para el sistema de recomendación.
- ❖ Validar las propuestas mediante pruebas de funcionalidad, seguridad y de carga y estrés.

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

- ❖ Descripción de las características del proveedor de servicios OAI-PMH desarrollado previamente.
- ❖ Realización de un estudio sobre el mecanismo de comunicación y forma de descripción de los documentos que establece el protocolo OAI-PMH.
- ❖ Realización de un estudio sobre las características y funcionalidades de los sistemas de recomendación.
- ❖ Selección de las tecnologías, herramientas, estándares y algoritmos a emplear en la implementación de la propuesta de solución, así como, la selección de la metodología de desarrollo.
- ❖ Definición de los requisitos funcionales y no funcionales de la propuesta de solución.
- ❖ Implementación de pruebas de funcionalidad, seguridad y carga y estrés.

A continuación se detallan los **métodos científicos** empleados durante la investigación.

Analítico-Sintético: para extraer los elementos más importantes que guían la investigación, realizando un análisis de toda la información acerca de las tecnologías, metodologías y herramientas posibles a ser utilizadas en el desarrollo del sistema, resumiendo sus características y analizando la viabilidad de cada una, para de esta forma integrarlas a una aplicación basada en la arquitectura del sistema de gestión de contenidos *Drupal*, en su versión 7.x que emplea OAI-PMH para el intercambio de la información.

Histórico-Lógico: método teórico que permite realizar un estudio profundo de la bibliografía existente acerca del desarrollo y evolución de los sistemas de recomendación, el uso de OAI-PMH en estos y las tecnologías que garantizan el correcto funcionamiento de este tipo de sistema.

Modelación: para modelar el comportamiento del sistema, tanto en el resultado esperado como en artefactos de ingeniería que describen el funcionamiento. También para modelar los flujos de datos, las

relaciones entre datos y roles de usuarios, de esta forma se desarrolla el sistema sobre la base de un resultado esperado.

El presente Trabajo de Diploma está compuesto por:

Capítulo I: en esta sección se realiza una caracterización del entorno conceptual y tecnológico del sistema de recomendación para un proveedor de servicios OAI-PMH. Se lleva a cabo un estudio sobre las características de los sistemas de recomendación enmarcando la investigación en los sistemas de recomendación basados en contenido, su surgimiento y posterior desarrollo. Además se realiza un análisis de las características de sistemas existentes semejantes al que se desea desarrollar y se hace referencia a los principales conceptos asociados al dominio del problema. También se caracteriza el entorno tecnológico y se define la metodología de desarrollo a emplear.

Capítulo II: en esta sección se realiza la propuesta de solución del sistema de recomendación para un proveedor de servicio OAI-PMH, para ello se representa el dominio del problema y se especifican los requisitos funcionales y no funcionales con los que debe cumplir el sistema para su correcto funcionamiento. Además se describen las funcionalidades del sistema, se especifican y analizan los casos de uso. También se hace una propuesta del diagrama de clases del diseño que posibilitan un mayor entendimiento del sistema a desarrollar.

Capítulo III: en esta sección se describe la implementación y se realizan las pruebas al sistema desarrollado además de presentarse los diagramas de componentes y despliegue.

Capítulo 1: Caracterización del entorno conceptual y tecnológico del sistema de recomendación para un proveedor de servicios OAI-PMH

Introducción

El presente capítulo hace referencia a los principales conceptos asociados al dominio del problema para que el usuario logre tener un mayor entendimiento sobre el tema a tratar. Además se brinda una información detallada sobre los principales sistemas de recomendación existentes y sus características. Se hace énfasis en los sistemas de recomendación basados en contenido. También se selecciona la metodología de desarrollo de *software* a emplear, las herramientas y tecnologías que se utilizar para el desarrollo de la aplicación y se aborda sobre las principales características de las mismas.

1.1. Conceptos asociados al dominio del problema

Para comprender los sistemas de recomendación, se hace necesario previamente definir algunos conceptos.

1.1.1. Preferencias

Según el autor *Mas-Colell* de la universidad de *New York* el concepto de preferencia surge de la modelación de las disyuntivas a las que se ven enfrentados los consumidores cuando tienen que elegir entre un objeto de consumo y otro. *Mas-Colell* afirma que como supuesto inicial se toma que el consumidor puede realizar una comparación entre dos o más objetos de manera ordinal, de esta comparación puede resultar: que el consumidor sea indiferente a los objetos que está considerando o que considere un objeto preferido por encima de otro (MAS-COLELL, 1995). Atendiendo al dominio del problema es válido aclarar que este concepto puede evidenciarse en las preferencias de los usuarios sobre determinados temas de investigación.

1.1.2. Recomendación

Una recomendación intenta proporcionar sugerencias significativas para los recursos que los usuarios puedan encontrar interesantes y útiles. Estas sugerencias se fundamentan en las calificaciones que un usuario le da a los recursos a través de una votación o mediante su historial de navegación. Como nota adicional se destaca que si un usuario no ha calificado ningún recurso o no cuenta con navegación alguna, el sistema no le podrá brindar una recomendación (KONSTAN, 2004). Es por esta razón que se proponen varias preferencias para los usuarios y de esta forma tener una mejor exactitud a la hora de realizar una

Capítulo 1: Caracterización del entorno conceptual y tecnológico del sistema de recomendación para un proveedor de servicios OAI-PMH

correcta recomendación al usuario.

1.1.3. Sistema de recomendación

Los sistemas de recomendación constituyen herramientas que tienen como principal objetivo asistir a los usuarios en sus procesos de búsquedas de información. De esta forma se filtran los *ítems* (término que se utiliza en evaluación para referirse a una pregunta en una prueba de corrección objetiva) (ALDERSON, 1995) de información haciendo uso de un grupo de recomendaciones. Estas son generadas a partir de las opiniones proporcionadas por los usuarios sobre esos *ítems* o mediante las preferencias de los mismos. Uno de los principales retos que tienen que afrontar los sistemas de recomendación es la gestión eficaz del enorme volumen de datos que almacenan, con el objetivo de facilitar a los consumidores la información que satisfaga sus necesidades de una manera rápida y sencilla. Esta necesidad obtiene una mayor importancia en una sociedad como la actual, donde el nivel de exigencia de los usuarios es cada vez mayor (KONSTAN, 2004).

1.2. Estructura de los sistemas de recomendación

Los sistemas de recomendación se encargan de pedir consejos o recomendaciones a expertos en la materia o seguir a aquellos individuos que tienen gustos similares al del usuario, o de lo contrario seleccionar objetos que tienen características similares a objetos que le hayan gustado anteriormente o que se parecen al que inicialmente buscaba (BROWN, 2012).

Los sistemas de recomendación se han ido consolidando como potentes herramientas para ayudar a reducir la sobrecarga de información a la que se enfrentan los internautas en los procesos de búsqueda de información y ayudan a filtrar los *ítems* de información que no son más que la división en parte de las preferencias o necesidades de un determinado usuario.

Existen varios elementos que intervienen en el esquema de funcionamiento de un sistema de recomendación: entradas y salidas del proceso de generación de la recomendación, el grado de personalización y los métodos usados para generar las recomendaciones (HERRERA, 2004).

A continuación se profundiza en el método usado para generar las recomendaciones el cual resalta por su importancia en el sistema.

1.2.1. Métodos de generación de recomendaciones

Los métodos de generación de recomendaciones que se emplean habitualmente no son mutuamente exclusivos entre sí, sino complementarios, o sea, que un mismo sistema de recomendación puede hacer uso de uno o varios de estos métodos. Las recomendaciones se generan a partir de las opiniones

Capítulo 1: Caracterización del entorno conceptual y tecnológico del sistema de recomendación para un proveedor de servicios OAI-PMH

proporcionadas por otros usuarios sobre esos *ítems* en búsquedas previas, o bien a partir del perfil del usuario. Un sistema de recomendación está asociado con un conjunto de *ítems* y su objetivo es recomendar a los usuarios *ítems* que les puedan ser de interés (FERRERA, 2011). La implementación de técnicas para el desarrollo de los sistemas de recomendación está íntimamente relacionada con el tipo de información que se vaya a utilizar. Una primera fuente de información a tener en cuenta es el tipo de *ítems* con los que se va a trabajar. Seguidamente se muestran algunos de los métodos de recomendación más importantes.

- ❖ **Recuperación pura o recomendación nula:** En este método de generación de recomendaciones el sistema ofrece a los usuarios una interfaz de búsqueda a través de la que pueden realizar consultas a una base de datos de *ítems*. Se trata de un sistema de búsqueda por lo que técnicamente no es un método de recomendación, aunque ante los usuarios aparece como tal (BROWN, 2012).
- ❖ **Recomendaciones seleccionadas manualmente:** Algunos sistemas utilizan este método generalmente realizado por expertos, como por ejemplo editores, artistas o críticos, en el caso de recomendaciones de películas o de música. Los expertos identifican *ítems* basándose en sus propias preferencias, intereses u objetivos, y crean una lista de estos *ítems* que estén disponible para todos los usuarios del sistema (BROWN, 2012).
- ❖ **Resúmenes estadísticos:** Algunos sistemas ofrecen estos resúmenes calculados en función de las opiniones del conjunto de usuarios, por lo que tampoco son personalizados. Por ejemplo, se podrían tener en cuenta el porcentaje de usuarios a los que ha satisfecho o han comprado un artículo, número de usuarios que recomiendan un *ítem*, o una evaluación media de todos los usuarios con respecto al *ítem* (Herrera-Viedma, 2012).

Para generar las recomendaciones hay dos posibilidades comúnmente aceptadas que dan lugar a dos grandes grupos de sistemas de recomendación, los colaborativos y los no colaborativos (HERRERA, 2004).

1.2.2. Clasificación de los sistemas de recomendación

Esencialmente todos los sistemas de recomendación tienen el mismo objetivo, guiar al usuario mediante recomendaciones a aquellos productos que más le puede interesar. Las técnicas utilizadas para producir estos objetivos difieren unas de otras significativamente, tanto en la información requerida como en los procesos necesarios para llevar a cabo estas recomendaciones. Basado en estos procesos, se pueden clasificar los sistemas de recomendación de la siguiente forma (FERRERA, 2011):

Capítulo 1: Caracterización del entorno conceptual y tecnológico del sistema de recomendación para un proveedor de servicios OAI-PMH

- ❖ **Basados en contenido:** este tipo de sistema genera recomendaciones al usuario teniendo en cuenta un *ítem* similar que él haya seleccionado.
- ❖ **Filtrado colaborativo:** se les llama también sistemas de recomendación sociales. Si a un conjunto de usuarios de un grupo le gusta un determinado *ítem* es de suponer que a los demás usuarios de ese grupo les guste también.
- ❖ **Recomendaciones basadas en conocimiento:** las sugerencias de sus *ítems* se basan en inferencias sobre las necesidades de los usuarios y sus preferencias. Para ello se utiliza conocimiento, donde se tiene información sobre cómo un *ítem* específico responde a una necesidad en particular del usuario, por lo tanto, la razón sobre la relación entre la necesidad y una posible recomendación.
- ❖ **Sistema de recomendación híbrido:** los sistemas de recomendación híbridos contemplan varios tipos de sistemas de recomendación, el más común es el filtro colaborativo y el sistema basado en contenido. Como consecuencia se disminuyen las desventajas de ambos y se aumentan los beneficios de la recomendación.
- ❖ **Sistemas de recomendación basados en utilidad:** realiza las recomendaciones basándose en el cálculo de la utilidad de cada objeto con respecto a un perfil de usuario.

1.2.3. Sistema de Recomendación Basado en Contenido

Luego del análisis de los sistemas, se decidió optar por los sistemas de recomendación basados en contenido ya que en el caso de DIMA el principal recurso con que se cuenta son los metadatos de los documentos, elementos que constituyen contenido y que son la base de la recomendación.

La ventaja de estos sistemas con respecto al resto, es que se realiza la recomendación sin necesidad de que el usuario haya tenido un historial determinado, lo cual favorece el proceso de recomendación a realizarse, ya que DIMA aún no cuenta con una gran popularidad debido a su poco tiempo de creación. Es importante resaltar que cada usuario puede seleccionar su o sus preferencias según su interés personal y recibirá recomendaciones relacionadas con estas.

Los sistemas de recomendación basados en contenidos, también presentan algunas limitantes, que se deben tener presente para la realización del trabajo. Las principales desventajas son:

- ❖ **Sobre especialización:** consiste en que las recomendaciones dadas al usuario tienden a ser muy similares ya que se basan en la misma información.
- ❖ **Dependencia de objetos:** es difícil extraer buena información de los objetos para realizar un cálculo confiable de similitud (FERRERA, 2011).

Capítulo 1: Caracterización del entorno conceptual y tecnológico del sistema de recomendación para un proveedor de servicios OAI-PMH

A pesar de basar la recomendación en el contenido, el perfil del usuario en el sistema juega un papel importante, porque mediante el perfil se recoge la información sobre las preferencias.

1.2.4. Perfil de usuario

El perfil del usuario es utilizado en los sistemas de recomendación basados en contenido para almacenar información referente al usuario. El tipo de información que se puede contener en el perfil del usuario es (FONT, 2009):

- ❖ **Preferencias del usuario:** contiene una relación de los tipos de *ítems* en que el usuario está interesado (o que no está interesado).
- ❖ **Historial del usuario:** se almacenan las interacciones que el usuario ha tenido con el sistema. Por ejemplo, la valoración de un *ítem*. Dicho aspecto es importante ya que el sistema puede utilizarlo para aprender gustos y preferencias del usuario.

1.3. Protocolo y Estándar de metadatos

El protocolo OAI-PMH es el que se emplea en DIMA para el intercambio de información. Este protocolo divide su funcionamiento en dos estructuras fundamentales: los proveedores de datos y los proveedores de servicios. Los primeros se encargan de exponer los metadatos de los documentos que almacenan y el segundo es el encargado de recolectar los metadatos, realizar un proceso de indexación y proponer al usuario una interfaz para la recuperación de información. De esta forma se pueden concentrar las publicaciones científicas de varias fuentes en un repositorio único, según los intereses de una institución.

En la actualidad los proveedores de datos se dividen en revistas científicas y repositorios institucionales, conocidos como el dorado y el verde del acceso abierto. Los proveedores de servicios por su parte están desarrollados en múltiples plataformas informáticas, donde sobresalen las aplicaciones sobre *Java* y *Python*. El protocolo OAI-PMH es el encargado de comunicar proveedores de datos y de servicio independientemente de la diversidad en tecnología. Para esto basa su comunicación en un mecanismo de preguntas y respuestas utilizando el protocolo HTTP. Una directiva importante de OAI-PMH es que exige el uso del estándar de metadatos *Dublin Core* (DC) para describir los documentos (FERNÁNDEZ, M. PRESENTADO EN UCIENCIA, 2012).

En resumen, el protocolo OAI-PMH proporciona un conjunto de servicios que permite la exposición y la recolección de metadatos del repositorio y se compone de seis peticiones que especifican el servicio que se necesita, estas son:

- ✓ **Identify:** para recuperar información de identidad del proveedor de datos.

Capítulo 1: Caracterización del entorno conceptual y tecnológico del sistema de recomendación para un proveedor de servicios OAI-PMH

- ✓ **ListIdentifiers:** para recuperar los encabezados de un registro del proveedor de datos.
- ✓ **ListRecords:** para capturar los registros completos del proveedor de datos.
- ✓ **ListSets:** para recuperar la estructura de colecciones del proveedor de datos.
- ✓ **ListMetadataFormats:** para obtener la lista de formatos de metadatos que se maneja en el proveedor de datos.
- ✓ **GetRecord:** para recuperar un registro individual del proveedor de datos.

1.4. Modelo Vectorial

El proceso de recomendación está muy ligado a la tarea de la recuperación de información y en muchos aspectos, entre ellos, el tratamiento que le da a los *ítems* de investigación, existe proximidad con los modelos de recuperación de búsqueda. En el presente trabajo se toma como base los principios del modelo vectorial para la recuperación de información.

El modelo vectorial es una evolución del modelo booleano, uno de los modelos primarios de recuperación de información. El modelo booleano está basado en la teoría de conjuntos y el álgebra booleana. En este modelo no existen las respuestas intermedias, aunque con algunos elementos de lógica difusa se trató de atenuar esa situación. El modelo vectorial propone un marco de trabajo en el que los pesos son valores no binarios para incluir respuestas intermedias. Los pesos de los términos se utilizan para obtener el grado de similitud entre los documentos y la respuesta (RODRÍGUEZ, J. A, 2004).

Conceptualmente, este modelo (vectorial) utiliza una matriz documento–término que contiene el vocabulario de la colección de referencia y los documentos existentes. En la intersección de un término t y un documento d se almacena un valor numérico de importancia del término t en el documento d ; tal valor representa su poder de discriminación. Así, cada documento puede ser visto como un vector que pertenece a un espacio n -dimensional, donde n es la cantidad de términos que componen el vocabulario de la colección. En teoría, los documentos que contengan términos similares estarán a muy poca distancia entre sí. De igual forma se trata a la consulta como un documento más. Luego, a partir de una consulta dada es posible devolver una lista de documentos ordenados por distancia (los más relevantes primeros). Para calcular la semejanza entre el vector consulta y los vectores que representan los documentos se utilizan diferentes fórmulas de distancia o similitud, siendo la más común la del coseno (BORDIGNON, 2007).

1.5. Medidas de similitud entre documentos

Las medidas de similitud permiten optimizar el sistema en función de la pregunta y la colección de documentos. Para el cálculo de similitud entre documentos según el modelo del espacio vectorial se encuentran: la función del coseno, el producto escalar, la fórmula de *Jaccard* y el coeficiente de Dice (TOLOSA, 2011).

En la presente investigación se profundiza en la función del coseno la cual fue seleccionada debido a que es la única que trabaja sobre vectores normalizados, por lo que elimina la incidencia que tenga la dimensión del vector. Por la naturaleza de los datos que se manejan en DIMA es muy común la comparación entre vectores de diferentes tamaños.

La función de similitud del coseno se basa en el coseno del ángulo entre dos documentos (TOLOSA, 2011). Si el ángulo es cercano a uno, entonces estos documentos son semejantes, si es cercano a cero son diferentes (CAMPOS, 2007). Partiendo de la representación vectorial de dos documentos $Q=(q_1, q_2, q_3, \dots, q_n)$ y $D=(d_1, d_2, d_3, \dots, d_n)$ en un espacio vectorial de N , donde q_i y d_i representan los pesos del término i -ésimo de los vectores Q y D respectivamente y además N es el número de términos distintos de la colección, entonces la función del coseno quedaría:

$$\frac{\sum_{i=1}^n q_i * d_i}{\sqrt{\sum_{i=1}^n (q_i)^2} * \sqrt{\sum_{i=1}^n (d_i)^2}} \quad (1)$$

1.6. Análisis de soluciones existentes

Hoy en día los sistemas de recomendación se han convertido en una herramienta clave para la selección de la preferencia de cada usuario y esto se debe al gran aporte que estos hacen para acercarse a las preferencias de cada persona. A continuación se muestran algunos ejemplos de estos sistemas, en este caso, los sistemas de recomendación basados en contenido ya que son los más idóneos para el desarrollo del sistema debido a sus características.

1.6.1. Referral Web

Es un sistema interactivo para la construcción, visualización y búsqueda de redes sociales en el *World Wide Web*. Este modela una red social mediante un grafo en cuyos nodos son representados los individuos y las conexiones entre nodos indican una relación directa descubierta entre ellos. Para identificar la relación directa entre individuos, se utilizan métodos tales como: referencias encontradas en las páginas personales, referencias de co-autores en publicaciones técnicas, citas en las publicaciones y

Capítulo 1: Caracterización del entorno conceptual y tecnológico del sistema de recomendación para un proveedor de servicios OAI-PMH

organigramas. *Referral Web* no intenta crear nuevas comunidades sino más bien ayudar a los usuarios a hacer un uso más eficiente de las redes existentes (HUITRÓN, 1997).

1.6.2. Fab

Es un sistema híbrido, basado en contenido y filtrado colaborativo que recomienda al usuario páginas *Web*. Combinando ambas técnicas *Fab* reduce las limitaciones de cada una de ellas. El sistema modela el perfil del usuario basado en el contenido de los análisis (cuando un usuario califica una página) y compara estos perfiles para determinar similitudes entre usuarios para una recomendación colaborativa. De esta manera el usuario recibirá páginas, tanto las que ha calificado relevantes (con respecto a su perfil) como las que han recibido calificaciones altas por usuarios con un perfil similar al suyo (vecinos cercanos). La arquitectura de *Fab* se encuentra formada por tres componentes principales: agentes de colección, que seleccionan páginas de un tema específico; agentes de selección, que encuentran páginas para un usuario específico y una herramienta central que los conecta. Cada agente mantiene un perfil del usuario basado en las palabras que contienen las páginas de *Web* que el usuario ha calificado (BROWN, 2012).

1.6.3. Diva

Es un sistema interactivo de vídeos que tiene como objetivo ofrecer al usuario una lista de películas basándose en su perfil y de esta forma evitar que busque por su cuenta a través de catálogos o motores de búsquedas. El reto para este sistema suele suceder, es cómo representar las preferencias del usuario y al mismo tiempo tener la flexibilidad necesaria con respecto a los cambios del sistema. A diferencia de otros sistemas, DIVA emplea algoritmos de aprendizaje que requieren de una serie de ejemplos previos para comenzar a generar una recomendación (BROWN, 2012).

1.6.4. SlideShare

Es un sistema que realiza una recomendación de forma híbrida. Incluye contenido y colaboración entre usuarios, además analiza los perfiles de usuario. Es un sistema muy popular en Internet, se comparte fundamentalmente información técnica. Hace un uso muy activo de las notificaciones vía correo electrónico.

1.6.5. ResearchGate

Sistema de intercambio de información científica. Tiene un funcionamiento muy similar al de una red social, pero la base del intercambio es la producción científica. Se generan comunidades de intereses y la recomendación no solo incluye documentos, también personas, instituciones y temas de investigación. El proceso de recomendación se basa en contenidos, en perfil de usuario, en un *ranking* (rg) que se crea para cada usuario y en principios de geolocalización. También hace un gran uso de las notificaciones vía correo

Capítulo 1: Caracterización del entorno conceptual y tecnológico del sistema de recomendación para un proveedor de servicios OAI-PMH

electrónico. Este sistema actualmente está siendo una fuente para obtener citas por parte de los investigadores, gracias a su buena indexación en grandes plataformas como *Google Scholar*.

1.6.6. Sistemas de recomendación desarrollados en la UCI

En la Universidad de las Ciencias Informáticas también se han desarrollado algunos sistemas de recomendación, basados en diversos principios y orientados a distintos escenarios. Se puede señalar el trabajo de (ZAMORA, 2013), donde se aplica la recomendación entre usuarios y no el análisis del contenido. Por su parte (SANTANA, 2013) realiza la recomendación de objetos de aprendizaje tomando como base el perfil de usuario y (FERIA, 2012) implementa un sistema colaborativo para la recomendación de videos. A pesar de las diferencias entre unos y otros, todos tienen en común la especial atención que prestan al usuario, ya sea mediante el análisis de su perfil o atendiendo a sus acciones en la aplicación.

1.7. Aportes del estudio de los sistemas de recomendación a la propuesta de solución

Los sistemas antes estudiados se analizaron con el objetivo de conocer cómo se gestiona toda la información relacionada con los sistemas de recomendación para de esta forma tener un conocimiento previo y así realizar una recomendación más cercana y apropiada al usuario. Del estudio podemos concluir que:

- El contenido es un elemento de análisis independientemente del tipo de sistema de recomendación.
- Siempre es conveniente tomar en cuenta los datos del perfil de usuario y sus acciones en el sistema.
- Las recomendaciones por sugerencias entre usuarios están sujetas a incertidumbre.
- Los principios de geolocalización y colaboración aunque se emplean en algunos sistemas no constituyen un elemento común.

1.8. Metodología de Desarrollo de Software

En un proyecto de desarrollo de *software* la metodología define *Quién* debe hacer, *Qué*, *Cuándo* y *Cómo* debe hacerlo. La metodología constituye la columna vertebral del proceso de desarrollo de *software* (RODRÍGUEZ, 2007).

El concepto de metodología de desarrollo ha sido objeto de estudio de varios autores, algunos de estos son:

Capítulo 1: Caracterización del entorno conceptual y tecnológico del sistema de recomendación para un proveedor de servicios OAI-PMH

“Se define metodología como un conjunto de filosofías, etapas, procedimientos, reglas, técnicas, herramientas, documentación y aspectos de formación para los desarrolladores de sistemas de información” (MADDISON, 1983).

Se llega a la definición de metodología de desarrollo como “un conjunto de procedimientos, técnicas, herramientas, y un soporte documental que ayuda a los desarrolladores a realizar nuevo software” (PIATTINI, 1996).

Cada metodología tiene características particulares, en este sistema se propone utilizar la metodología ágil *OpenUP*, ya que se ajusta al sistema, está diseñada para entornos dinámicos y para equipos de desarrollo pequeños.

OpenUP es un Proceso Unificado que aplica enfoques iterativos e incrementales dentro de un ciclo de vida estructurado, utiliza una filosofía ágil que se enfoca en la naturaleza de colaboración en el desarrollo de *software*. El objetivo de *OpenUP* es ayudar al equipo de desarrollo, a lo largo de todo el ciclo de vida de las iteraciones, para que sea capaz de añadir valor de negocio a los clientes, de una forma predecible, con la entrega de un *software* operativo y funcional al final de cada iteración (RÍOS SALGADO, 2013).

El uso de la metodología ágil *OpenUP* favorece el desarrollo del sistema de recomendación ya que:

- ❖ Permite que el sistema sea fácilmente adaptable a los requerimientos cambiantes que se dieron durante el desarrollo del proyecto.
- ❖ Proporciona un marco de referencia para dividir el proyecto en etapas.
- ❖ Contribuye en la generación de la documentación del proyecto.
- ❖ Permite realizar una planificación real.

Los resultados obtenidos del estudio realizado demuestran que la metodología *OpenUP* contribuye eficientemente al cumplimiento de todas las especificaciones funcionales y de rendimiento establecidas para el sistema. Finalmente, por la experiencia adquirida, se puede afirmar que *OpenUP*, contribuye a que la aplicación tenga características que todo buen *software* debe poseer tales como: eficiencia, flexibilidad, portabilidad e integridad (RÍOS SALGADO, 2013).

1.9. Herramientas CASE

Las herramientas *CASE* (*Computer Aided Software Engineering*) son diversas aplicaciones informáticas destinadas a aumentar la productividad en el desarrollo de *software* reduciendo el costo en términos de tiempo y dinero (TERRY, 2002). Estas herramientas permiten ayudar en todos los aspectos del ciclo de

Capítulo 1: Caracterización del entorno conceptual y tecnológico del sistema de recomendación para un proveedor de servicios OAI-PMH

vida de desarrollo del *software* en tareas como el proceso de realizar el diseño del proyecto, cálculo de costos, compilación automática, documentación o detección de errores. En resumen, las herramientas *case* son un conjunto de herramientas y ayudas que facilitan a los ingenieros de *software* y analistas, el desarrollo del mismo de principio a fin o en alguna de sus fases.

1.9.1. Visual Paradigm

Visual Paradingm es una herramienta de diseño que permite usar el Lenguaje de Modelado Unificado (UML), está diseñado para apoyar al desarrollo de sistemas dentro de la rama de la ciencia de la computación. Soporta estándares tales como *UML*, *SysML*, *BPMN*, *XMI*, entre otros. Ofrece un conjunto completo de herramientas de desarrollo necesario para la captura de requisitos, la planificación y pruebas, el modelado de clases y datos.

1.10. UML Lenguaje Unificado de Modelado

Es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema. Brinda un estándar para representar un plano del sistema, agregando aspectos conceptuales tales como: procesos del negocio, funciones del sistema, expresiones de lenguajes de programación, esquemas de base de datos y componentes de software reutilizables (VALDÉS, mayo 2010).

1.11. La pila tecnológica de DIMA

El sistema de recomendación a desarrollar debe funcionar dentro de una aplicación ya existente, en este caso DIMA. Esta aplicación está desarrollada tomando como base el sistema de gestión de contenidos (CMS, por sus siglas en inglés: *Content Managent System*) *Drupal 7.23*. Las tecnologías que se empleen para el proceso de implementación y despliegue deben coincidir con las que actualmente soportan el desempeño de DIMA.

1.11.1. MySQL

Los *SGBD* ofrecen un control centralizado de la información teniendo como objetivos evitar la redundancia de los datos, mejorar los mecanismos de seguridad de los mismos y la privacidad, mantener la integridad de los datos realizando las validaciones necesarias y mejorar la eficacia del acceso a los datos.

En el caso de MySQL, es un SGBD relacional multi-hilo y multiusuario, es muy común verlo relacionado con aplicaciones desarrolladas en lenguaje PHP (*Hypertext Pre-processor*). Líderes de la industria como *Yahoo*, *Google*, *Nokia* y *Youtube* lo utilizan. Se ofrece bajo licencia GNU GPL (*GNU General*

Capítulo 1: Caracterización del entorno conceptual y tecnológico del sistema de recomendación para un proveedor de servicios OAI-PMH

Public License), mientras que aquellas empresas que quieran incorporarlo en productos privativos deben de comprar una licencia que les permita su uso. Cuantiosos CMS como *Drupal*, *Joomla* y *phpBB* usan *MySQL* (SUÁREZ ROMERO, 2012).

Algunas de sus ventajas:

- ✓ **Coste:** el coste es gratuito para la mayor parte de los usos.
- ✓ **Portabilidad:** se ejecuta en la gran mayoría de los sistemas operativos y en la mayor parte de los casos los datos se pueden trasladar de un sistema a otro sin presentar dificultad alguna.
- ✓ **Facilidad de uso:** es un sistema fácil de utilizar y de administrar. Las herramientas de *MySQL* son fuertes y flexibles, sin sacrificar su capacidad de uso.

1.11.2. Servidor Web Apache 2.2.21

Apache es un servidor altamente configurable de diseño modular libre y de código abierto. Es multiplataforma. Posee una alta integración con diversas tecnologías, lenguajes, plataformas, bases de datos, entre otros. Opera con lenguajes como *Java*, *Perl* y *PHP*. Tiene capacidad para servir páginas tanto de contenido estático como de contenido dinámico a través de otras herramientas soportadas que facilitan la actualización de los contenidos mediante bases de datos, ficheros u otras fuentes de información.

1.11.3. Lenguaje de programación PHP 5.3.8

Es un lenguaje de código abierto que se interpreta con un alto nivel embebido en páginas *HTML* y se ejecuta en el servidor al nivel más básico, *PHP* puede procesar la información de formularios, generar páginas con contenidos dinámicos, o mandar y recibir *cookies*. Las 4 grandes características de este lenguaje de programación son: velocidad, estabilidad, seguridad y simplicidad. Este lenguaje presenta una gran documentación y excelente comunidad.

Este lenguaje interpretado y multiplataforma de propósito general es ampliamente usado, se encuentra instalado en más de veinte millones de sitios *Web* y más de un millón de servidores. El mismo permite la conexión a diferentes tipos de servidores de bases de datos tales como *MYSQL*, *PostgreSQL*, *Oracle*, *ODBC*, *DB2*, *Microsoft SQL Server*, *Firebird* y *SQLite*. De igual forma, posee soporte para la Programación Orientada a Objetos, soporte para *Unicode* (*PHP* 6), manejo de excepciones, entre otros (SECURITY.PHP, 2010).

Capítulo 1: Caracterización del entorno conceptual y tecnológico del sistema de recomendación para un proveedor de servicios OAI-PMH

1.11.4. HTML

HTML (del inglés: *HiperText Markup Language*) es un lenguaje de marcado diseñado para estructurar textos y presentarlos en forma de hipertextos, que es el formato estándar de las páginas *Web*. Fue creado por *Tim Berners-Lee* en 1986.

HTML presenta diversas ventajas dentro de las cuales se pueden encontrar las siguientes:

- ✓ No necesita de grandes conocimientos cuando se cuenta con un editor de páginas *Web*.
- ✓ Archivos pequeños.
- ✓ Despliegue rápido.
- ✓ Lenguaje de fácil aprendizaje.
- ✓ Lo admiten todos los navegadores *Web*.

En este trabajo se hace uso de *HTML* versión 5, entre otros aspectos porque permite reducir el código en hojas de estilo y en ficheros js. Además incluye ventajas para la presentación de grandes volúmenes de datos, para el manejo de formularios y para la presentación de documentos RDF (*Resource Description Framework*).

1.11.5. CSS 3

CSS es un lenguaje de hojas de estilos creado para controlar la presentación de los documentos electrónicos definidos con *HTML* y *XHTML*. Es la mejor forma de separar los contenidos y su presentación y es imprescindible para la creación de páginas *Web* complejas. La separación de los contenidos y su presentación presenta numerosas ventajas, ya que obliga a crear documentos *HTML/XHTML* bien definidos y con significado completo (también llamados “documentos semánticos”). Además, mejora la accesibilidad del documento, reduce la complejidad de su mantenimiento y permite visualizar el mismo documento en infinidad de dispositivos diferentes (EGUÍLUZ PÉREZ, 2011).

A continuación se describen algunas ventajas de usar *CSS*:

- ✓ Separación del contenido y presentación.
- ✓ Flexibilidad.
- ✓ Optimización de los tiempos de carga y de tráfico en el servidor.
- ✓ Precisión o elasticidad.
- ✓ Accesibilidad y estructuración.
- ✓ Limpieza del código fuente.
- ✓ Estandarización frente a especificaciones propietarias.
- ✓ Permite la diferenciación de estilos para imprimir / visualizar en pantalla.

1.11.6. Drupal

Drupal es un sistema gestor de contenidos que tiene una amplia comunidad de usuarios que lo utilizan y lo mantienen, lo que significa que la aplicación tiene un desarrollo dinámico y que se está mejorando continuamente. Su código fuente está disponible bajo los términos de la licencia *GNU/GPL* motivo por el cual es posible extenderlo o adaptarlo según las necesidades, al contrario de otros *CMS* propietarios. Al estar estructurado en módulos proporciona una gran flexibilidad, permitiendo crear desde un portal *Web* de un instituto hasta un sitio de *blogs*.

Puede decirse que desde sus inicios *Drupal* es multiplataforma ya que funciona con *Apache* o *Microsoft IIS* como servidor *Web* y en sistemas operativos como *Linux*, *Solaris*, *Windows*, *Mac OS X* el código generado es muy limpio, cumpliendo con los estándares de accesibilidad del consorcio internacional que produce recomendaciones en todo el mundo de la *Web* (GONZÁLEZ, 2011).

La principal ventaja de *Drupal* es su flexibilidad y su arquitectura basada en entidades desde la versión 7.x, que le permite el crecimiento y la adaptación a diferentes entornos. También es importante destacar su avanzado motor de plantillas y la cantidad de módulos contribuidos.

1.12. Entorno de Desarrollo Integrado (IDE)

Un IDE es una aplicación compuesta por un conjunto de herramientas útiles para un desarrollador. Puede ser exclusivo para un lenguaje de programación o bien, poder utilizarse para varios. Suele consistir de un editor de código (con facilidades como resaltado de sintaxis, completamiento de código y navegación entre clases), un compilador y herramientas de automatización de la compilación, un depurador y en algunos casos un constructor de interfaz gráfica. Dentro de los más populares a nivel mundial se encuentran el Eclipse y el *Microsoft® Visual Studio®* (ALEGSA.ALEGSA.COM.AR., 2010).

1.12.1. NetBeans

Es un IDE modular normalizado, escrito en el lenguaje de programación *Java*. El proyecto *NetBeans* consiste en un IDE de código abierto y una plataforma de aplicación, las cuales pueden ser usadas como una estructura de soporte general para compilar cualquier tipo de aplicación (HERNÁNDEZ, 2013).

En su versión 7.1, bajo licencia GPL y de código abierto esta herramienta tiene la finalidad de permitirle a los desarrolladores crear diferentes sistemas y proyectos orientados sobre todo a la creación de soluciones en lenguaje *Java*, ya sea que se encuentren en *Java SE* (Edición Estándar) o *Java EE* (Edición Empresarial), además de soportar otros lenguajes tales como *PHP* y *Javascript*.

1.13 JMeter Apache 4.0

Herramienta *Java* dentro del proyecto de *Jakarta*, que permite realizar pruebas de rendimiento y pruebas funcionales sobre aplicaciones web y bases de datos. *JMeter* se destaca por su versatilidad, estabilidad, por ser de uso gratuito, permite realizar pruebas web clásicas, incluye test de FTP, JDBC, JNDI, LDAP, SOAP/XML-RPC y *Web Service* (en Beta). También permite la ejecución de pruebas distribuidas entre distintos ordenadores para realizar pruebas de rendimiento (GARCÍA, 2012).

1.14 Acunetix Web Vulnerability Scanner

Potente utilidad que analiza una página Web en busca de posibles vulnerabilidades. Detecta técnicas de hacking como, por ejemplo, inyección SQL, ataques de ejecución de código y ataques de autenticación, entre otros. Detecta posibles agujeros de seguridad en páginas Web y otros dispositivos clientes (ANTUNES, 2009).

Capítulo 2: Propuesta de solución del sistema de recomendación para un proveedor de servicio OAI-PMH.

Introducción

En el presente capítulo, se realiza un análisis de las principales entidades que tributan al funcionamiento del sistema y se especifican los requisitos funcionales y no funcionales con los que debe cumplir la aplicación. También se muestran los diagramas de casos de uso del sistema, el modelo de dominio y el modelo de diseño.

2.1. Objeto de automatización

El objeto de automatización parte de la existencia de un conjunto de documentos, cada uno de estos descritos por los quince metadatos que propone el estándar *Dublin Core*. Se gestionan las preferencias, la asociación de documentos y usuarios con preferencias, además de la comparación entre documentos y preferencias empleando los metadatos título, descripción y palabras claves.

2.2. Propuesta del sistema

El sistema realiza la indexación de documentos desde proveedores de datos OAI, basándose en peticiones OAI. Primeramente el sistema debe mostrar un formulario donde el usuario pueda autenticarse dejando constancia de su estancia por el sitio para así ofrecerle una mejor recomendación. Una vez que el usuario esté autenticado en el sistema este debe mostrar un bloque que contenga las categorías que representan a los documentos, el usuario selecciona una o varias preferencias y el sistema muestra documentos asociados a su elección. El sistema contiene otras funciones importante como desclasificar documento, clasificar documentos de forma automática y clasificar documentos de forma manual, las cuales solo van a ser ejecutadas por el administrador o aquel usuario que tenga los permisos necesarios.

Para desclasificar un documento se mostrará una vista que muestra todos los documentos clasificados existentes en el sistema, luego se seleccionan aquellos documentos a desclasificar y se pulsa el botón guardar. La opción que clasifica los documentos automáticamente, muestra una vista que brinda la posibilidad de clasificar automáticamente 10, 50, 100 o todos los documentos existentes en la base de datos. Por su parte la función clasificadora de documentos manuales luego que el administrador o usuario con permiso escoge la categoría a la que desea que pertenezca este documento, lo marca y presiona el

botón guardar para que este documento cambie de categoría.

2.3. Modelo de Dominio

Un modelo de dominio es una representación visual estática del entorno real objeto proyecto, que captura los tipos más importantes de objetos en el contexto del sistema. Los objetos representan las cosas que existen o los eventos que suceden en el entorno en que trabaja el sistema. Además se describe mediante diagramas UML (especialmente mediante diagramas de clases). Son diagramas donde se representan los objetos del dominio o eventos y las relaciones que pueden ocurrir entre ellos (HIDALGO, 2008).

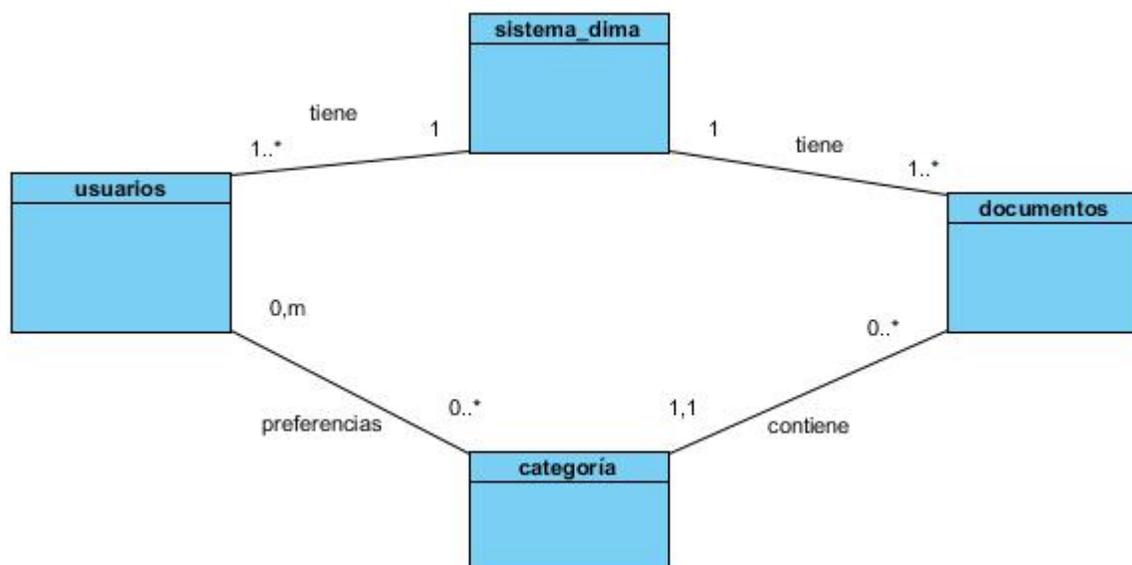


Figura 1: Modelo de dominio

2.3.1. Descripción de las clases del modelo de dominio

| Nombre de la clase | Descripción |
|--------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Sistema DIMA | Proporciona las acciones de los usuarios autenticados en el sistema y les brinda a estos la posibilidad de obtener documentos clasificados relacionados con sus preferencias. Tiene uno o varios usuarios instanciados y uno o varios documentos clasificados. |
| Usuario | Realiza la selección de aquellas categorías de su interés. |
| Categoría | Contiene documentos clasificados. |

| | |
|------------|-----------------------------------|
| Documentos | Contiene uno o muchas categorías. |
|------------|-----------------------------------|

Tabla 1. Descripción de las clases del modelo de dominio

2.4. Requisitos

La Ingeniería de Requisitos cumple un papel primordial en el proceso de producción de *software*, debido a que va enfocada a un área fundamental en el desarrollo de todo sistema, que es la definición de lo que se desea producir. Su principal tarea consiste en la generación de especificaciones correctas que describan con claridad, sin ambigüedades, en forma consistente y compacta el comportamiento del sistema, de manera que le garantice minimizar los problemas relacionados al desarrollo de sistemas. Los requisitos se encuentran divididos en 2 grandes grupos: los funcionales y los no funcionales (FERRERA, 2011). A continuación la representación de estos:

2.4.1. Requisitos Funcionales

Los requisitos funcionales representan las necesidades del cliente reflejadas en el sistema y expresan la naturaleza de su funcionamiento. O sea, se concentran en describir cómo interacciona el sistema con su entorno el cuál va a ser su estado y funcionamiento en un momento dado, aunque en ocasiones es factible indicar también qué no deberá realizar el sistema (BROWN, 2012).

Seguidamente se presentan los requisitos funcionales por los que deberá estar guiado el desarrollo del sistema.

✓ **RF1 Autenticar usuario**

El sistema debe mostrar un formulario de autenticación donde el usuario pueda realizar la acción.

✓ **RF2 Gestionar perfil de usuario**

El sistema debe permitir gestionar los datos asociados a los usuarios y guardar las nuevas preferencias respecto a los documentos disponibles en él.

✓ **RF3 Gestionar clasificación**

El sistema debe permitir agregar y eliminar documentos a una preferencia para que estas sirvan de base a la clasificación.

✓ **RF4 Gestionar documento**

El sistema debe ser capaz de permitirle al usuario administrador o usuario con permisos realizar la gestión de los documentos.

✓ RF5 Mostrar recomendación

El sistema debe ser capaz de mostrar una vista donde se le recomiende documentos al usuario.

2.5. Requisitos No Funcionales

Luego de analizar las condiciones que resultan apropiadas para el funcionamiento de la propuesta de solución planteada, se identificaron los Requisitos No Funcionales (RNF) que no son más que las propiedades o cualidades que el producto debe tener. En general son aquellas características que hacen que el producto sea atractivo, usable, rápido y/o confiable. Estos están muy vinculados a los requisitos funcionales, debido a que una vez que se conozca lo que el sistema debe realizar, se puede determinar cómo este ha de comportarse en determinadas circunstancias, qué cualidades debe tener y cuán rápido debe ser (BROWN, 2012).

Seguidamente los requisitos no funcionales que debe cumplir el sistema para realizar una correcta recomendación.

2.5.1. Usabilidad

RNF 1. El sistema debe poseer una arquitectura simple y que posibilite a los usuarios llegar al contenido que desea en un corto tiempo.

RNF 2. El sistema debe permitirles a los usuarios con pocos conocimientos de informática como poder interactuar con el sistema.

RNF 3. El sistema debe poseer una interfaz intuitiva y fácil de navegar.

2.5.2. Disponibilidad

RNF 4. El sistema debe estar disponible las 24 horas del día los 7 días de la semana.

2.5.3. Confiabilidad

RNF 5. Las recomendaciones deben ser fiables.

RNF 6. La información no podrá ser modificada por ningún usuario no autorizado, protegiendo así la integridad de los datos.

2.5.4. Fiabilidad

RNF 7. El sistema debe mostrarle al usuario un mensaje indicándole que ha ocurrido un fallo en la operación que se realice erróneamente.

2.5.5. Legales

RNF 8. El sistema se basa en Drupal, por lo que su esquema de licenciamiento debe ser GNU/GPL.

RNF 9. Documentación legal de uso como Declaración de Autoría.

2.5.6. Soporte

RNF 10. El sistema debe dar la posibilidad de ser mejorado, así como de incorporarle nuevas funcionalidades, en caso de ser necesarias.

2.5.7. Hardware

✓ Servidores de Aplicación y BD:

RNF 11. Para garantizar un funcionamiento óptimo del sistema, el servidor donde estará desplegado el mismo deberá tener características mayores o semejantes a: un procesador *Corei5*, con una velocidad del procesador de 2.5 GHz, 8 GB de RAM y 160 GB de disco duro. El servidor de base de datos debe ser *Corei3*, con 4GB de RAM y 500 GB de disco duro.

✓ PC cliente:

RNF 12. Para el cliente como requerimientos mínimos: Procesador *Pentium III* a 2.8 GHz con 512 Mb de memoria RAM y una tarjeta de red.

✓ Software:

RNF 13. Para acceder a la aplicación se requiere usar un navegador Web: *Mozilla Firefox 15* o superior, *Opera*, *Safari 5.1* o superior o *Internet Explorer 8* o superior.

✓ Servidores de Aplicación:

RNF 14. Para un servidor de aplicación Web se requiere usar Apache versión 2.0 o superior con módulo PHP 5.0 v5.4.3 disponible, este debe estar configurado con las extensiones “*pdo*” y “*mysql*”; y un servidor de base de datos *MySQL 5.5.16* o superior.

✓ Servidores de BD:

RNF 15. Sistema Gestor de base de datos *MySQL 5.5.16*

✓ Seguridad

RNF 16. El sistema le permitirá al usuario el acceso a las funcionales en dependencia del rol que posean.

RNF 17. Se utilizará el protocolo *HTTPS* para la comunicación entre el cliente y el servidor en los procesos de envío de datos entrados por el usuario en la autenticación y en las tareas administrativas y de gestión de contenidos.

✓ **Portabilidad:**

RNF 18. El sistema será multiplataforma, pudiendo ser implantado en sistemas operativos como Linux y Windows así como en sus respectivas distribuciones, teniendo un correcto y eficaz funcionamiento sin importar la plataforma en la que se utilice.

2.6. Modelo de casos de uso

El modelo de casos de uso describe las funcionalidades propuestas del nuevo sistema. Un caso de uso representa una unidad discreta de interacción entre un usuario (humano o máquina) y el sistema. Un caso de uso es una unidad simple de trabajo significativo.

Cada caso de uso tiene una descripción que describe la funcionalidad que se construirá en el sistema propuesto. Un caso de uso puede "incluir" la funcionalidad de otro caso de uso o "extender" a otro caso de uso con su propio comportamiento (S.A., 2013).

En esta sección se identifican los posibles actores del sistema a desarrollar, y se definen los casos de uso del sistema.

2.6.1. Diagrama de casos de uso

Los casos de uso constituyen fragmentos de funcionalidades que el sistema debe ofrecer para lograr aportar un resultado de valor a sus actores. De manera un poco más precisa un caso de uso especifica una secuencia de acciones que el sistema puede llevar a cabo interactuando con sus actores, incluyendo alternativas dentro de la secuencia. El diagrama de casos de uso del sistema constituye una representación gráfica de los procesos y su interacción con los actores. Se utiliza para ilustrar los requisitos del sistema al mostrar cómo reacciona una respuesta a eventos que se producen en el mismo (JACOBSON, 2000).

Seguidamente se representan el diagrama de casos de uso del sistema propuesto para este ciclo de desarrollo, plasmándose todos los casos de uso necesarios para satisfacer los requisitos funcionales identificados.

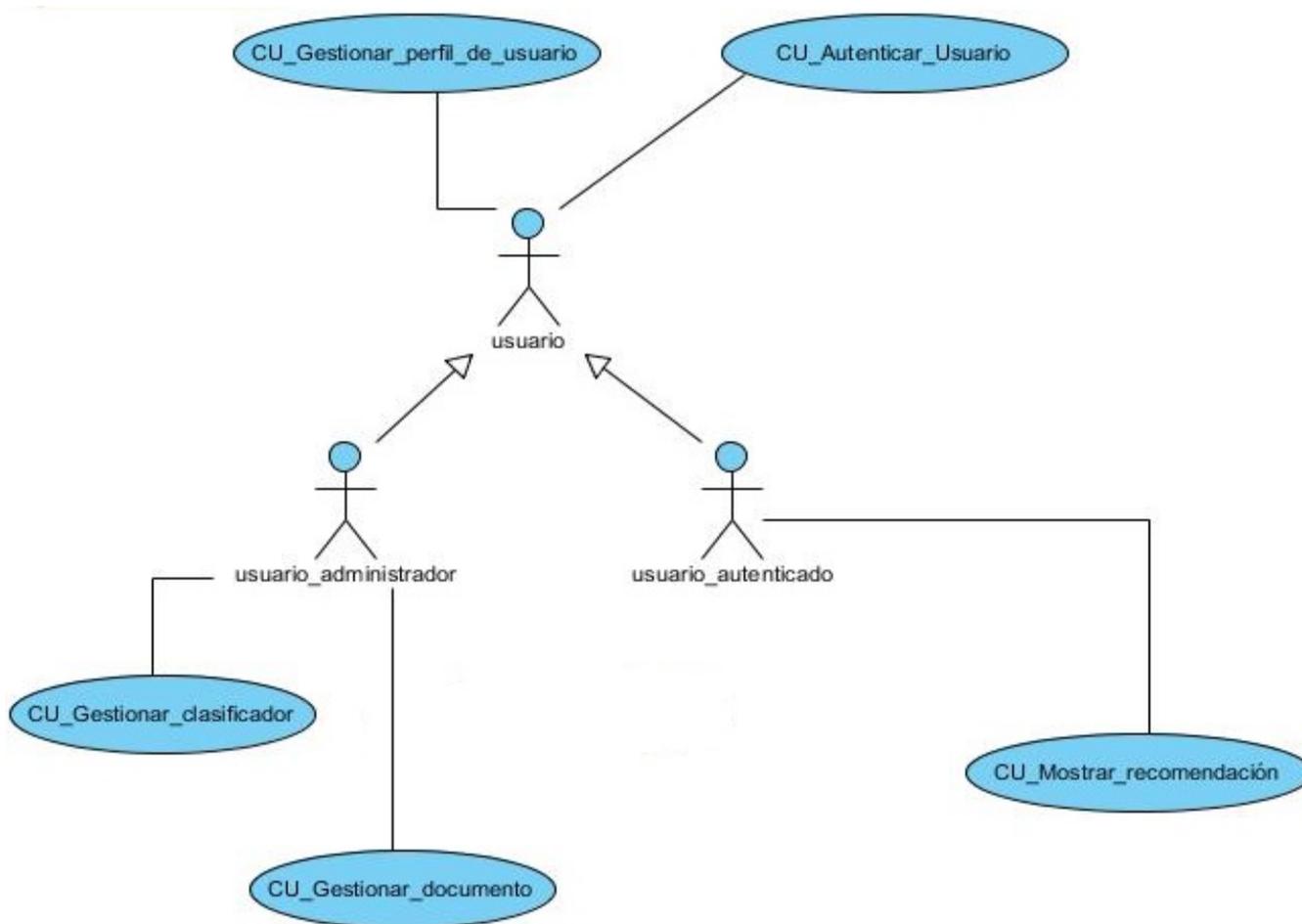


Figura 2: Diagrama de Casos de Usos del Sistema

2.6.2. Descripción de los casos de uso

| | |
|------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Caso de uso | Autenticar Usuario |
| Objetivo | Permitir la identificación del usuario en el sistema. |
| Actores | Usuario |
| Resumen | El CU se inicia cuando un usuario desea ingresar al sistema, el sistema muestra la interfaz de autenticación que solicita las credenciales, finaliza así el CU. |
| Complejidad | Alta |
| Prioridad | Crítico |
| Precondiciones | - |
| Postcondiciones | Se obtuvo la información de acuerdo al criterio de autenticación introducido. |

Capítulo 2: Propuesta de solución del sistema de recomendación para un proveedor de servicio OAI-PMH

| Flujo básico Autenticar Usuario | | |
|----------------------------------------------|------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| | Actor | Sistema |
| 1. | Indica en el buscador la dirección del sistema. | |
| 2. | | El sistema muestra la interfaz de autenticación. |
| 3. | El usuario elige la opción cuenta y realiza su autenticación, luego oprime el botón Iniciar sección. | |
| 4. | | El sistema comprueba que el nombre de usuario y la contraseña son correctos. Si son correctos automáticamente se le da el acceso a este. |
| 5. | | Termina el CU. |
| 1ª Criterio de autenticación erróneo. | | |
| | Actor | Sistema |
| 2.1 | | Si la cuenta introducida no es la correcta el sistema muestra las mismas casillas pero esta vez de color rojo como muestra que su cuenta es errónea y nuevamente le brinda la posibilidad al usuario de intentar la autenticación. |

Tabla 2. Descripción del caso de uso Autenticar usuario

| Caso de uso | Gestionar perfil de usuario |
|-----------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Objetivo | Permitir gestionar los datos asociados a los usuarios y guardar las nuevas preferencias respecto a los documentos disponibles en el sistema. |
| Actores | Usuario |
| Resumen | El CU se inicia cuando un usuario autenticado desea gestionar sus preferencias o perfil. El sistema permite al usuario seleccionar sus preferencias documentales que luego servirán de referencia para realizar las recomendaciones. |
| Complejidad | Media |
| Prioridad | Crítico |
| Precondiciones | Estar autenticado. |

Capítulo 2: Propuesta de solución del sistema de recomendación para un proveedor de servicio OAI-PMH

| | | |
|-----------------------------------|---------------------------------------------------------------------------|----------------------------------------------------------------------------------------------|
| Postcondiciones | Las preferencias del usuario quedan registradas. | |
| Flujo de eventos | | |
| Flujo básico editar perfil | | |
| | Actor | Sistema |
| 1. | Selecciona el campo editar. | |
| 2. | Dentro de la opción Preferencias elige la o las preferencias de su gusto. | |
| 3. | Presiona el botón guardar. | |
| 4. | | El sistema guarda estas preferencias para luego mostrarle documentos relacionados con estas. |
| | | Termina el CU. |

Tabla 3. Descripción del caso de uso Gestionar perfil de usuario

| | | |
|-----------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------|
| Caso de uso | Gestionar clasificador | |
| Objetivo | Permitir clasificar documentos de forma automática. | |
| Actores | Usuario administrador | |
| Resumen | El CU comienza cuando el usuario administrador desea que el sistema clasifique documentos de forma automática. El sistema brinda la posibilidad de definir cuantos documentos se desean clasificar y procede a realizar la clasificación automática basado en los documentos antes clasificados. | |
| Complejidad | Alta | |
| Prioridad | Crítico | |
| Precondiciones | El usuario debe tener el permiso requerido y deben existir documentos ya clasificados. | |
| Postcondiciones | Se asocia un documento a una preferencia. | |
| Flujo de eventos | | |
| Flujo básico: clasificación automática | | |
| | Actor | Sistema |
| 1 | Selecciona en el menú de configuración el área de flujo de trabajo. | |

Capítulo 2: Propuesta de solución del sistema de recomendación para un proveedor de servicio OAI-PMH

| | | |
|---|--------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------|
| 2 | Escoge la opción clasificación automática. | |
| 3 | | Muestra cuantos documentos se desean clasificar, las opciones son: 10, 50, 100 o todos. |
| 4 | Selecciona la cantidad de documentos a clasificar y presiona el botón guardar. | |
| 5 | | Se realiza la clasificación automática y se guardan las configuraciones. Termina el CU. |

Tabla 4. Descripción del caso de uso Gestionar clasificador

| | | |
|-------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------|
| Caso de uso | Mostrar recomendación | |
| Objetivo | Mostrar una vista de recomendaciones al usuario de posibles documentos que sean de su interés basándose en sus preferencias. | |
| Actores | Usuario autenticado | |
| Resumen | El CU se inicia una vez que el usuario autenticado ha seleccionado sus preferencias documentales. El sistema analiza esta información y le muestra una vista con los documentos que cree serían de su interés. | |
| Complejidad | Alta | |
| Prioridad | Crítico | |
| Precondiciones | El usuario debe estar autenticado y tener preferencias registradas. | |
| Postcondiciones | El usuario recibe una recomendación. | |
| Flujo de eventos | | |
| Flujo básico Mostrar recomendación | | |
| | Actor | Sistema |
| 1. | Va a la página recomendaciones. | |
| 2. | | El sistema muestra una interfaz con los documentos que pudieran ser del interés del usuario en dependencia de las categorías escogidas. Termina el CU. |

Tabla 5. Descripción del caso de uso Mostrar recomendación

| | |
|--------------------|----------------------------|
| Caso de uso | Gestionar documento |
|--------------------|----------------------------|

Capítulo 2: Propuesta de solución del sistema de recomendación para un proveedor de servicio OAI-PMH

| | | |
|------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------|
| Objetivo | Gestionar la relación entre documentos y preferencias que sirva de base para el proceso de clasificación. | |
| Actores | Usuario administrador | |
| Resumen | El CU se inicia una vez que el administrador decide incorporar un documento a una preferencia o desligar un documento de una preferencia. El sistema permite marcar el documento y seleccionar la preferencia o directamente después de marcar el documento da la opción de desclasificar. | |
| Complejidad | Alta | |
| Prioridad | Crítico | |
| Precondiciones | El usuario debe estar autenticado. | |
| Postcondiciones | Se asociar documentos y preferencias. | |
| Flujo de eventos | | |
| Flujo básico Clasificación manual | | |
| | Actor | Sistema |
| 1. | Selecciona la opción clasificación manual del menú flujo de trabajo dentro del área de configuraciones. | |
| 2. | | Muestra a lista de preferencias y de documentos sin clasificar. |
| 3. | Selecciona documentos y preferencia. | |
| 4. | | Asigna los documentos marcados a la preferencia seleccionada. Termina el CU. |
| Flujo básico Desclasificar | | |
| | Actor | Sistema |
| 1. | Selecciona la opción desclasificar documento del menú flujo de trabajo dentro del área de configuraciones. | |
| 2. | | Muestra la lista de documentos clasificados por preferencias. |
| 3. | Selecciona los documentos que desea clasificar. | |

| | |
|----|----------------------------------------------------------------------|
| 4. | Elimina la relación del documento con la preferencia. Termina el CU. |
|----|----------------------------------------------------------------------|

Tabla 6. Descripción del caso de uso Gestionar documento

2.7. Estilos arquitectónicos

Los estilos arquitectónicos son un conjunto de reglas de diseño que identifican las clases de componentes y conectores que se pueden utilizar para componer un sistema o subsistema, junto con las restricciones locales o globales de la forma en que la composición se lleva a cabo. Los componentes, incluyendo los subsistemas encapsulados, se pueden distinguir por la naturaleza de su computación (LARMAN, 1999). Los estilos arquitectónicos se agrupan en varias familias. Seguidamente se abordan los elementos principales del estilo que se utilizará en esta investigación.

Estilos de Llamada y Retorno: esta familia enfatiza en los cambios a sistemas o aplicaciones. Son los estilos más generalizados en sistemas a gran escala. Los miembros de esta familia son las arquitecturas de programa principal y subrutina, los sistemas basados en llamadas a procedimientos remotos, los sistemas orientados a objetos y los sistemas jerárquicos en capas. Se mencionan a continuación las arquitecturas que están dentro de este tipo de estilo debido a su importancia en el desarrollo de sistemas (LARMAN, 1999).

- ❖ Arquitectura Modelo Vista Controlador.
- ❖ Arquitectura en Capas.
- ❖ Arquitectura Orientada a Objetos.
- ❖ Arquitectura Basada en Componentes.

2.7.1. Patrones arquitectónicos

Los patrones arquitectónicos o patrones de arquitectura ofrecen soluciones a problemas de arquitectura de *software*. Estos brindan una descripción de los elementos y el tipo de relación que tienen, junto con un conjunto de restricciones sobre cómo pueden ser usados. Un patrón arquitectónico expresa un esquema de organización estructural esencial para un sistema de *software*, que consta de subsistemas, sus responsabilidades e interrelaciones. En comparación con los patrones de diseño, los patrones arquitectónicos tienen un nivel de abstracción mayor. Por otra parte dichos patrones conforman el nivel en el que la arquitectura de *software* define la estructura básica de un sistema, pudiendo estar relacionado con otros patrones y representa una plantilla de construcción que provee un conjunto de subsistemas, aportando las normas para su organización (BROWN, 2012).

Capítulo 2: Propuesta de solución del sistema de recomendación para un proveedor de servicio OAI-PMH

Para el desarrollo del módulo se hace uso del CMS *Drupal* en el cual se decide utilizar el patrón arquitectónico **Modelo – Vista – Controlador (MVC)**. Este es un patrón de arquitectura de *software* encargado de separar la lógica de negocio de la interfaz del usuario y es uno de los más utilizados en aplicaciones *Web*. Este facilita la funcionalidad, sostenibilidad y escalabilidad del sistema, de forma simple y sencilla. El patrón MVC divide las aplicaciones en tres niveles de abstracción (BROWN, 2012):

Modelo: representa la lógica de negocios. Es el encargado de acceder de forma directa a los datos, actuando como intermediario con la base de datos.

Vista: es la encargada de mostrar la información al usuario de forma gráfica y humanamente legible.

Controlador: es el intermediario entre la vista y el modelo. Es quien controla las interacciones del usuario solicitando los datos al modelo y entregándolos a la vista para que ésta, lo presente al usuario, de forma humanamente legible.

En este patrón tanto la vista como el controlador dependen del modelo, el cual no depende de las otras clases. Esta separación permite construir y probar el modelo independientemente de la representación visual. La separación entre vista y controlador puede ser secundaria en aplicaciones de clientes ricos, de hecho, muchos *frameworks* de interfaz implementan ambos roles en un solo objeto (REYNOSO, 2004).



Figura 3: Funcionamiento del patrón de arquitectónico MVC

2.8. Patrón de Diseño

Los patrones de diseño representan un mecanismo utilizado para describir soluciones generales a problemas de diseño que pueden ser reutilizadas en la construcción de una aplicación. Cada patrón de diseño prescribe una estructura de clase, sus roles y colaboraciones y una adecuada asignación de métodos para resolver un problema de diseño de una manera flexible y adaptable (BROWN, 2012).

Los patrones de diseño se encuentran divididos en dos grandes grupos, el primero serían los Patrones Generales de Asignación de Responsabilidad de *Software* (GRASP) que establecen los principios fundamentales de la asignación de responsabilidades a objetos y el segundo los llamados *Gang of Four* o Banda de los Cuatro (GOF) que se clasifican en tres grupos (estructurales, creacionales y comportamiento). Los estructurales tratan la combinación de clases, su relación y la formación de estructuras de alta complejidad, los creacionales manejan la creación de instancias y los de comportamientos la interacción y la cooperación entre clases (FERRERA, 2011).

Seguidamente se describen los patrones de diseño utilizados por el *CMS Drupal* y utilizados en la implementación del sistema de recomendación:

2.8.1. Patrones GOF

- **Instancia única (*Singleton*):** patrón creacional a nivel de objetos. Su propósito es garantizar que una clase sólo tenga una única instancia, proporcionando un punto de acceso global a la misma (FERRERA, 2011). En general, estos objetos no encapsulan datos, lo que separa a un módulo de otro es el conjunto de funciones que contiene, por lo que debe ser pensado como una clase con una sola instancia.

Un ejemplo de la utilización de este patrón se evidencia en la creación de las categorías y la clasificación de los documentos donde estos se complementan a la hora de realizar la recomendación pero se implementan de forma independiente.

- **Decorador (*Decorator*):** patrón estructural a nivel de objetos. Su propósito es añadir responsabilidades adicionales a un objeto dinámicamente, proporcionando una alternativa flexible a la especialización mediante herencia, cuando se trata de añadir funcionalidades (FERRERA, 2011).

La utilización de este patrón de diseño se evidencia en el archivo *docrecomender.admin.inc* en el que se tiene la función *docrecomender_settings_manual* (). Esta función devuelve una matriz que contiene los tipos de categorías y las características de los documentos, donde luego esta función va a ser llamada desde el archivo *docrecomender.module* para brindarle al usuario documentos relacionados con estas categorías.

- **Observador (*Observer*):** define una dependencia del tipo uno a muchos entre objetos, de manera que cuando uno de los objetos cambia su estado, el observador se encarga de notificar este cambio a todos los otros dependientes (MOLINA, 2005). El uso de este patrón se evidencia principalmente en la realización de la recomendación al usuario por parte del sistema. Para poder recomendar un documento, se necesita saber si el usuario seleccionó alguna preferencia. Para ello se implementa la función *docrecommender_form_user_profile_form_alter (&\$form,&\$form_state)* la cual Drupal la tiene definida, se modifica el formulario de usuario y se agregan las preferencias; luego se genera una lista *checkbox* que es donde se permite marcar las preferencias por el usuario.
- **Puente (*Bridge*):** es utilizado para desacoplar una abstracción de su implementación, de manera que ambas puedan ser modificadas independientemente sin necesidad de alterar por ello la otra. Tanto la abstracción como su implementación deben ser extensibles por subclases (MOLINA, 2005). Este patrón se evidencia en la creación del módulo cuando se forma el sistema de recomendación, donde los archivos que este contiene pueden ser modificados independientemente sin que la funcionalidad del otro se vea afectada.
- **Acción (*Command*):** patrón de comportamiento a nivel de objetos. Su propósito es encapsular en un objeto la acción que satisface una petición, permitiendo ejecutar dicha operación sin necesidad de conocer el contenido de la misma (FERRERA, 2011). Ejemplo de la utilización de este patrón se puede apreciar dentro del propio módulo creado, donde se implementan funciones que solo implementan los ganchos (funciones que le indican al núcleo de *Drupal* que operación va a realizar y sobre que módulo la hará) necesarios para su funcionamiento, en este caso se puede citar el ejemplo de la *function docrecommender_schema()* la cual modifica el esquema de bases de datos creando las tablas compuestas por categoría, id, nombre y documento, id y categoría.

2.8.2. Patrones GRASP

- **Patrón Experto:** tiene como objetivo principal asignar una responsabilidad al experto en información (MOLINA, 2005). Se evidencia dentro del módulo *docrecommender* que se encuentran implementado en el patrón arquitectónico MVC (Modelo-Vista-Controlador). En este modelo a cada archivo se le asigna una tarea determinada según la responsabilidad que este posee en el funcionamiento del módulo, ya sea, mostrando la información correspondiente, accediendo a la base de datos del sistema o gestionando las operaciones a realizar. Ejemplo de ello se evidencia en la función *docrecommender_settings_manual_submit(\$form, &\$form_state)* la cual procesa el formulario de clasificación manual donde básicamente su funcionamiento se enviará al identificador del documento que se encuentra marcado en ese momento y al identificador que se encuentra marcado en el *checkbox*.

- **Creador:** tiene como objetivo principal asignar a las clases usadas responsabilidades de crear instancias de otras clases, soportando mayor claridad en el código y posibilitando el encapsulamiento y la reusabilidad (MOLINA, 2005). Un claro ejemplo de la utilización de este patrón se refleja dentro del módulo en el archivo *docrecomemder* en la función *docrecomender_settings()*, la cual procesa el formulario del algoritmo que realiza el proceso de clasificación de documentos el cual crea instancias de otros archivos del módulo para poder acceder a la información que se necesita para realizar la recomendación.
- **Bajo acoplamiento:** tiene como principal objetivo asignar una responsabilidad para mantener el bajo acoplamiento, o sea, mantener las clases lo menos ligadas posibles. Soporta el diseño de clases más independientes, que reducen el impacto de los cambios y también más reutilizables, acrecentando la oportunidad de una mayor productividad (BROWN, 2012). El uso de este patrón se encuentra reflejado en el módulo claramente debido a que el nivel de dependencias entre los ficheros pertenecientes al mismo módulo es muy bajo ya que en cada fichero implementado por el módulo se crea una instancia de los ficheros que necesite. De manera tal que cuando se produzca alguna modificación en alguno de ellos la repercusión en el otro sea mínima.

2.9. Modelo de diseño

El modelo de diseño es un modelo de objetos que describe la realización física de los casos de uso del sistema, centrándose, principalmente en el impacto que tienen los requisitos funcionales y no funcionales. Este modelo sirve de abstracción entre la implementación y el sistema y por ello es utilizado como una entrada fundamental de las actividades de implementación (JACOBSON, 2000).

2.9.1. Diagramas de clases del diseño

El diagrama de clases de diseño describe gráficamente las especificaciones de las clases de *software*, las interfaces en una aplicación, así como también sus relaciones (FERRERA, 2011). A continuación se muestra el diagrama de clases de diseño con estereotipos *Web*.

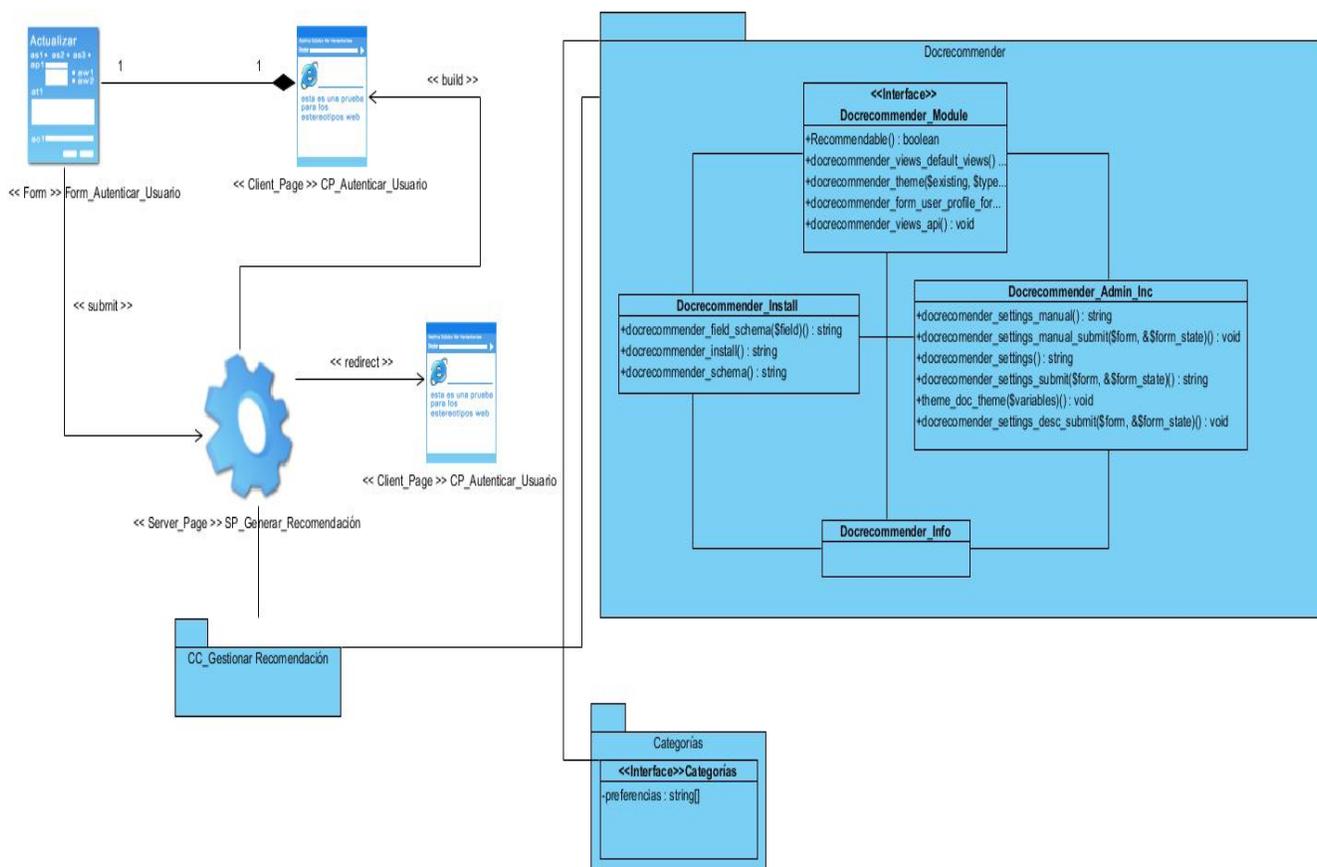


Figura 4: Diagrama de clases del diseño Generar Recomendación

2.9.2. Descripción de las clases del diagrama de clases de diseño

- ❖ **SP_Generar_Recomendación:** esta clase es la encargada de generar todo el flujo de trabajo del sistema. En ella se crean todos los elementos del menú y es la encargada de construir las vistas que serán mostradas al usuario.
- ❖ **CP_Autenticar_Usuario:** tiene la tarea de verificar los datos recogidos en los formularios.
- ❖ **Form_Autenticar_Usuario:** formulario que le permite al usuario autenticarse para de esta forma dejar constancia de su estancia por el sitio y así poder hacerle una mejor recomendación.
- ❖ **CP_Mostrar_Recomendación:** esta clase es la encargada de mostrar una vista donde se le recomiende documentos al usuario.
- ❖ **CC_Gestionar_Recomendación:** esta clase es la encargada de gestionar las acciones que se ejecutan en el módulo *docrecomemder*.

Capítulo 3: Implementación y validación de los resultados del sistema de recomendación para un proveedor de servicios OIA-PMH

Introducción

En el presente capítulo se describe el modelo de implementación, el cual está compuesto por el diagrama de componentes y de despliegue, para lograr esto se partió de los resultados obtenidos en el modelo de diseño descrito en el capítulo anterior y además se realizan los diferentes tipos de pruebas mencionadas en los objetivos específicos propuestos en la introducción.

3.1. Modelo de Implementación

El modelo de implementación describe cómo los elementos del modelo de diseño se implementan en términos de componentes. Describe también cómo se organizan los componentes de acuerdo con los mecanismos de estructuración y modularización disponibles en el entorno de implementación y en el lenguaje o lenguajes de programación utilizados y cómo dependen los componentes unos de otros (FERRERA, 2011). Seguidamente se muestra una vista general del modelo de implementación del sistema de recomendación propuesto para un proveedor de servicios OAI – PMH.

3.1.1. Diagrama de componentes

Los diagramas de componentes describen los elementos físicos del sistema y sus relaciones, representan todos los elementos que entran en la fabricación de aplicaciones informáticas; son usados para estructurar el modelo de implementación en términos de subsistemas de implementación y mostrar la relación entre ellos. Es un grafo de componentes unidos a través de relaciones que pueden ser de ejecución o compilación (SUÁREZ, 2012). El diagrama de componentes del sistema se representa a continuación:

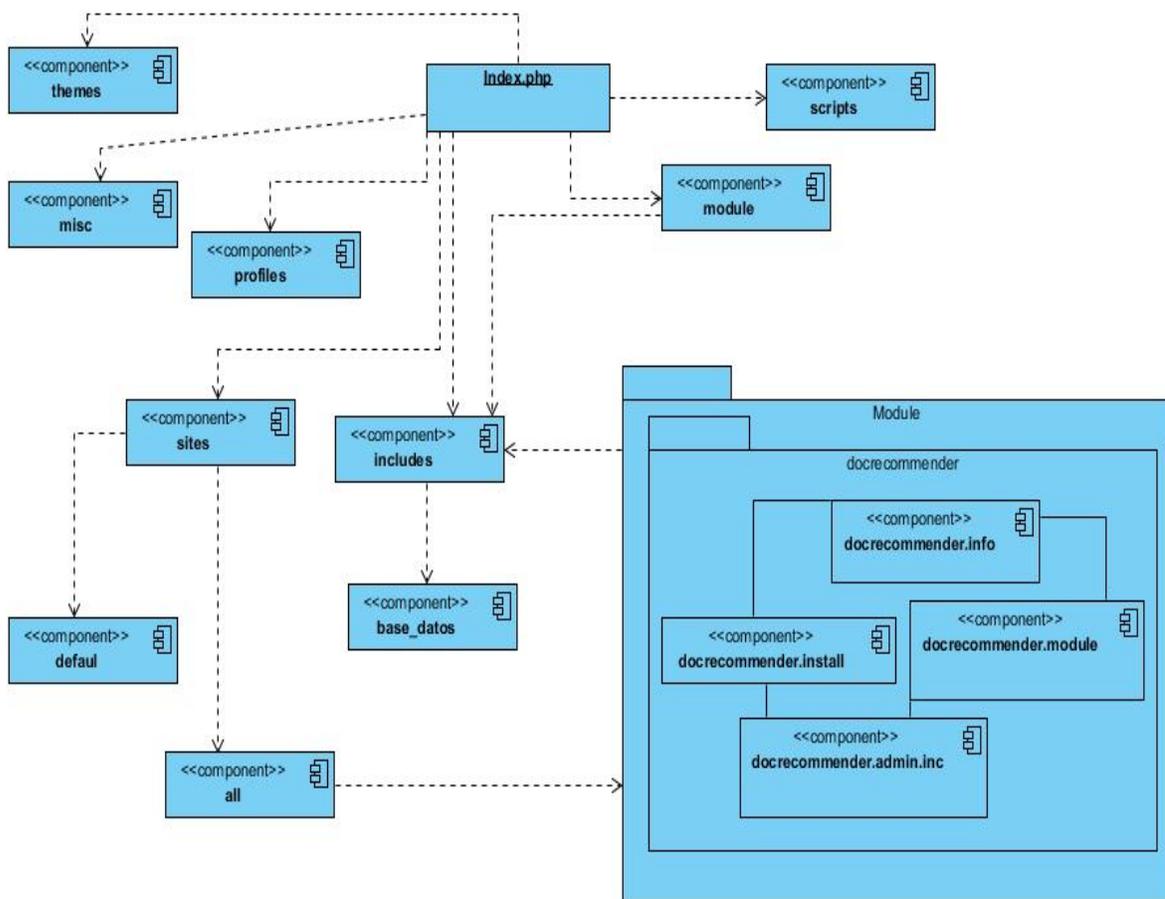


Figura 5: Diagrama de componentes

3.1.2 Descripción de los componentes del diagrama

| Componente | Descripción |
|----------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>index.php</i> | El archivo <i>index.php</i> es la puerta de entrada al sistema. Cuando cargamos una página del sitio web estamos siempre haciendo una llamada a <i>index.php</i> . |
| <i>themes</i> | La carpeta <i>themes</i> contiene los temas que vienen con la distribución de Drupal. Los temas nuevos que queramos añadir, crear o modificar, se deben subir a <i>/sites/all/themes</i> o <i>/sites/default/themes</i> . |
| <i>module</i> | La carpeta <i>modules</i> contiene los módulos del núcleo, cada uno en su carpeta correspondiente. Nunca se deben modificar directamente los módulos, y tampoco se deben subir en esta carpeta otros módulos adicionales. |
| <i>docrecomender</i> | Archivo donde se encuentran los ficheros que componen el módulo. |

Capítulo 3: Implementación y validación de los recursos del sistema de recomendación para un proveedor de servicio OAI-PMH

| | |
|--------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>docrecomender.info</i> | Fichero que almacena la información principal del módulo. |
| <i>docrecomender.install</i> | Fichero que alberga datos que se insertan en el sistema como estructura de los contenidos una vez que se instala el módulo. |
| <i>docrecomender.module</i> | Fichero que contiene el código del módulo implementado. |
| <i>includes</i> | La carpeta <i>includes</i> contiene un conjunto de librerías en forma de archivos PHP con extensión <i>.inc</i> , que incluyen funciones comunes del sistema como; <i>ajax.inc</i> , <i>batch.inc</i> , <i>cache.inc</i> , <i>date.inc</i> , y <i>form.inc</i> . |
| base de datos | Representa la base de datos donde se almacenan los documentos que son archivados. |
| <i>docrecomender.admin.inc</i> | Archivo que contiene las funciones que son llamadas desde el archivo <i>.module</i> . |
| <i>misc</i> | La carpeta <i>misc</i> contiene archivos javascript e imágenes requeridas por el sistema como <i>favicon.ico</i> y <i>jquery.js</i> . |
| <i>profiles</i> | La carpeta <i>profiles</i> contiene los perfiles de instalación. Inicialmente Drupal puede ser instalado en modo estándar (<i>standard</i>) o en versión reducida (mínimal), con los módulos mínimos requeridos para el funcionamiento de Drupal. |
| <i>sites</i> | La carpeta <i>sites</i> contiene modificaciones que se añaden a la distribución original además de contener módulos adicionales creados, añadidos o descargados del repositorio de módulos de <i>Drupal</i> , colocados en <i>sites/all/modules</i> . Además los temas adicionales instalados o creados, en <i>sites/all/themes</i> . La carpeta incluye tras la instalación el archivo de configuración del sitio (<i>settings.php</i>). |
| <i>scripts</i> | La carpeta <i>scripts</i> contiene utilidades adicionales que no utiliza <i>Drupal</i> directamente, pero se puede utilizar desde la línea de comandos de <i>shell</i> . Por ejemplo, el <i>script password-hash.sh</i> permite obtener una contraseña codificada a partir de la contraseña original (en texto plano) (RODRÍGUEZ, 2014). |

Tabla 7. Descripción de los componentes

3.2. Modelo de Despliegue

Un diagrama de despliegue muestra las relaciones físicas entre los componentes *hardware* y *software* en el sistema final, es decir, la configuración de los elementos de procesamiento en tiempo de ejecución y los

Capítulo 3: Implementación y validación de los recursos del sistema de recomendación para un proveedor de servicio OAI-PMH

componentes *software*; procesos y objetos que se ejecutan en ellos. Un diagrama de despliegue es un grafo de nodos unidos por conexiones de comunicación que describe la arquitectura física del sistema. Para poder realizar el despliegue del sistema en la institución que se vaya a beneficiar de él, la misma debe contar con una red LAN y PC clientes que posean un navegador *Web*. También es necesario un servidor de aplicación, y un servidor de Base de Datos. La comunicación entre las PC clientes y el servidor de aplicación en este caso es mediante el protocolo HTTPS (SOSA, 2009).

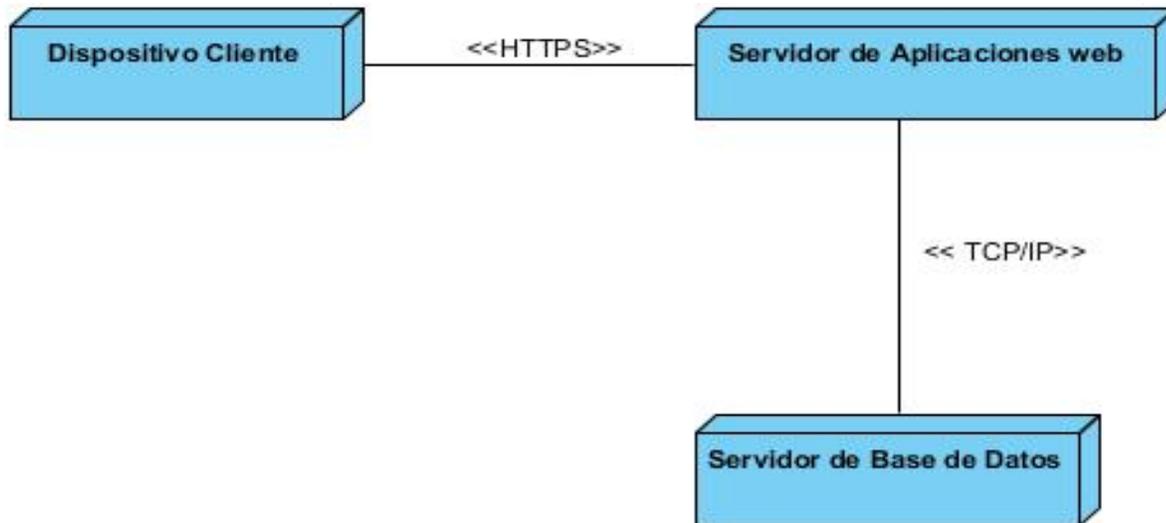


Figura 6: Modelo de despliegue

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

- ❖ **Dispositivo Cliente:** representa la computadora personal del cliente que se conecta al Servidor de Aplicaciones *Web*. La misma se comunica al servidor mediante el Protocolo de Transferencia de Hipertexto *HTTPS*.
- ❖ **Servidor de Aplicaciones *Web*:** representa el servidor sobre el cual se encuentra corriendo la aplicación *Web*. Este accede al Servidor de Base de Datos para el manejo de información.
- ❖ **Servidor de Base de Datos:** es la representación de la base de datos perteneciente a la aplicación (SUÁREZ, 2012).

3.3. Funciones importantes

A continuación se mostrarán las funciones que más sobresalen en la aplicación debido a su importancia y aporte a la misma.

3.3.1 Función para alterar el formulario de registro

La función *function form_user_profile_form_alter* modifica el formulario permitiendo agregar además de las funcionalidades que están definidas por *Drupal* generar una lista *checkbox* que le permita al sistema tener un previo conocimiento del interés del usuario para posteriormente mostrar documentos asociados con la o las preferencias del usuario.

```
function docrecommender_form_user_profile_form_alter(&$form,
&$form_state){
  global $user;
  $node = user_load($user->uid);
  $categories = array(array(1,'Bases de Datos'),array(2,'Ingeniería de
software'),
    array(3,'Lenguaje de
Programación'),array(4,'Internet'),array(5,'Sistemas Operativos'),
    array(6,'Simulación'),array(7,'Teleinformática'),array(8,'Redes'));

  $form['colours'] = array(
    '#type' => 'checkboxes',
    '#title' => t('Preferencias'),
    '#default_value' => variable_get('user_' . $node->name . '_color',array('page')),
    '#options' => array(
      '1' => 'Bases de Datos',
      '2' => 'Ingeniería de software',
      '3' => 'Lenguaje de Programación',
      '4' => 'Internet',
      '5' => 'Sistemas Operativos',
      '6' => 'Simulación',
      '7' => 'Tele-Informática',
      '8' => 'Redes'),
    '#description' => t('Selecciones las áreas de su interés.'));
  $form['#submit'][] = 'form_user_profile_alter_submit';
}
```

Figura 7: Función *form_user_profile_form_alter*

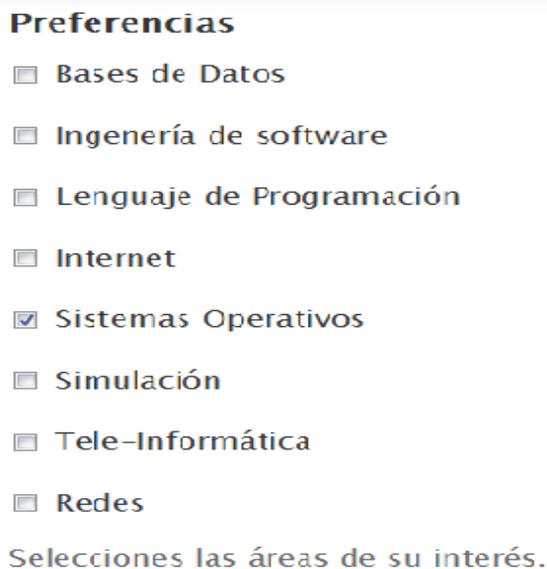


Figura 8: Selección de preferencias del usuario

3.3.2 Función para generar el menú de administración

La función *docrecommender_menu()* básicamente se dedica a registrar direcciones *HTTPS* y crear los tres menús del área del flujo de trabajo en la sección de configuración, para el funcionamiento del administrador o usuarios con permisos exclusivos. Dentro contiene el área de clasificación automática, clasificación manual y el área de desclasificar documento, cada una con su funcionamiento independiente. En la siguiente imagen se muestra un ejemplo del resultado de la misma.



Figura 9: Muestra de la vista del flujo de trabajo

3.3.3 Función para el envío del formulario de configuración

La función *Function_docrecomender_settings_submit()* es la encargada de realizar la clasificación de documentos mediante el modelo vectorial. Realiza los cálculos que permiten introducir un orden (*ranking*) en los documentos clasificados además de determinar el nivel de pertenencia de cierto documento a cierta categoría.

```
function docrecomender_settings_submit($form, &$form_state){

$count = 0;
if($form_state['values']['count'] == 'All')
    $count = 1000;
else
    $count = (int)$form_state['values']['count'];
    //Node of test 943
    //-----The query
    $query = db_query('SELECT n.nid FROM {node} n
WHERE n.nid NOT IN (SELECT m.nid FROM {document_category} m) AND type=\'document\' LIMIT ' . $count);

$find_not_doc = $query->fetchAll();
$query = db_query('SELECT * FROM {doc_categories}');
$find_all_categories = $query->fetchAll();
foreach($find_not_doc as $doc) {
    $consulta = array();
    $node = node_load($doc->nid);

    if(isset($node->title) && $node->title != null){
        $title = $node->title;
        $aux = explode(' ', $title);
        foreach($aux as $a){
            if(strlen($a) > 3){
                $consulta[] = strtoupper($a);
            }
        }
    }
}
}
```

Figura 10: Función *docrecomender_settings_submi()*

A continuación se cita un ejemplo donde el usuario se interesa por la opción de sistemas operativos y el sistema muestra las recomendaciones asociadas a estos.



| Titulo | Semejanza |
|--------------------------------------------------------------------------------------------------------------------------|-----------|
| Evaluaciones frecuentes en inglés en la asignatura Sistemas Operativos Modernos | Default |
| APLICACIONES DE SEGURIDAD INFORMÁTICA EN SISTEMAS OPERATIVOS LIBRES PARA LOGRAR LA SOBERANÍA TECNOLÓGICA | Default |
| APLICACIONES DE SEGURIDAD INFORMÁTICA EN SISTEMAS OPERATIVOS LIBRES PARA LOGRAR LA SOBERANÍA TECNOLÓGICA | Default |
| CONSTRUCCIÓN DE SISTEMAS OPERATIVOS BASADOS EN LINUX CON BUILDROOT | Default |
| CONSTRUCCIÓN DE SISTEMAS OPERATIVOS BASADOS EN LINUX CON BUILDROOT | Default |

Directorio de Artículos de Acceso Abierto | REDUNIV © 2013

Figura 11: Imagen que muestra las recomendaciones ofrecidas por el sistema una vez el usuario elegir su o sus preferencias

3.4 Pruebas del sistema

En el proceso de desarrollo de *software* es de suma importancia verificar la calidad y el adecuado funcionamiento del *software*, de ahí la existencia del proceso de pruebas que es tan o más importante que todas las realizadas hasta el momento, puesto que en ella se refleja la calidad con que ha sido llevada a cabo la construcción del sistema. En este proceso se ejecutan pruebas dirigidas a componentes del *software* o al sistema de *software* en su totalidad, teniendo como objetivo principal para medir el grado en que el sistema cumple con los requisitos, o sea, comprobar que el producto se comporta como desea el cliente. Para que las pruebas tengan éxito es necesario utilizar técnicas que guíen el proceso. Existen dos técnicas fundamentales en el proceso de pruebas que pueden realizarse en base a dos enfoques principales, estas son: las pruebas de caja blanca y las pruebas de caja negra (PRESSMAN, 2007).

- ❖ **Prueba de Caja Blanca:** en la prueba de caja blanca, se analiza la estructura lógica del programa y para cada alternativa que pueda presentarse, los datos de prueba ideados conducirán a ella.
- ❖ **Prueba de Caja Negra:** en la prueba de la caja negra, los casos de prueba pretenden demostrar que las funciones del *software* son operativas, que la entrada se acepta de forma adecuada y que se produce una salida correcta.

Luego de haber analizado la composición de la aplicación se decide aplicar pruebas de caja negra al módulo *Docrecommender*, con el objetivo de verificar la entrada de valores válidos y no válidos, permitiendo comprobar si el sistema cumple con el resultado esperado.

Capítulo 3: Implementación y validación de los recursos del sistema de recomendación para un proveedor de servicio OAI-PMH

Existen varias clasificaciones y tipos de pruebas, que se determinan según las características de calidad definidas por la norma ISO 9126 entre las que se encuentran (FEBLES ESTRADA, 2011):

❖ **Funcionalidad**

- ✓ Pruebas de funcionalidad
- ✓ Pruebas de seguridad
- ✓ Pruebas de volumen

❖ **Confiabilidad**

- ✓ Pruebas de recuperación y tolerancia a fallos
- ✓ Pruebas de Comparativa

❖ **Usabilidad**

- ✓ Pruebas de usabilidad
- ✓ Pruebas de estructura

❖ **Eficiencia**

- ✓ Pruebas de contenido
- ✓ Pruebas de carga
- ✓ Pruebas de estrés

❖ **Pruebas de rendimiento**

- ✓ Portabilidad
- ✓ Pruebas de configuración
- ✓ Pruebas de instalación

En la validación del sistema de recomendación para un proveedor de servicios OAI-PMH es necesario tener en cuenta varias características dependiendo del tipo de aplicación, además del objetivo que persiguen las pruebas a realizar. Después de analizar las propiedades del sistema (DIMA), se determinó que es necesario medir su reacción integral frente a diversas acciones que podrán efectuar los usuarios, para ello es conveniente la selección de las siguientes pruebas: pruebas de funcionalidad, de seguridad y de cargas y estrés. En el próximo epígrafe se describirán las pruebas seleccionadas y los resultados arrojados con su ejecución.

3.4.1 Pruebas funcionales

Las pruebas funcionales están centradas en comprobar que las funcionalidades descritas en el documento de requisitos del sistema se cumplen con la implementación realizada. A este tipo de pruebas también se les denomina pruebas de comportamiento o de caja negra, debido a que los analistas enfocan su atención a las respuestas del sistema de acuerdo a los datos de entrada y sus resultados en los datos de salida, los cuales se definen generalmente en los casos de prueba que se crean antes del inicio de las pruebas (ORÉ B, 2009).

El proceso para ejecutar este tipo de pruebas es el siguiente (SERNA M, 2011):

- ✓ Analizar los requisitos y sus especificaciones.
- ✓ Seleccionar entradas válidas y no válidas de acuerdo con las especificaciones.
- ✓ Determinar las salidas esperadas para cada entrada.
- ✓ Diseñar los casos de pruebas con las entradas seleccionadas.
- ✓ Ejecutar los casos de prueba.
- ✓ Comparar las salidas encontradas con las salidas esperadas.
- ✓ Determinar si el funcionamiento del *software* en prueba es apropiado.

A continuación se muestra el diseño de casos de pruebas correspondiente a las funcionalidades Gestionar perfil de usuario, Gestionar documento y Gestionar clasificador las cuales resaltan por su importancia en la aplicación e influencia en la misma.

| Escenario | Descripción | Preferencias | Respuesta del sistema | Flujo central |
|---------------------------------------------------|--------------------------------------------------------------------------|---------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| EC1 Gestionar perfil de usuario correctamente. | El sistema debe permitirle al usuario gestionar su perfil correctamente. | V Base de Datos, Internet | Permite gestionar los datos asociados a los usuarios y guarda las nuevas preferencias respecto a los documentos disponibles en el sistema. | 1- El usuario autenticado desea gestionar sus preferencias o perfil. 2- El sistema permite al usuario seleccionar sus preferencias que luego servirán de referencia para realizar las recomendaciones. |

Capítulo 3: Implementación y validación de los recursos del sistema de recomendación para un proveedor de servicio OAI-PMH

| | | | |
|-------------------------------------------------------|--------------------------------------------------------------------------|-----------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| EC 2 Gestionar perfil de usuario incorrectamente . | El sistema no le permite al usuario gestionar su perfil incorrectamente. | l (Vacío), no se realice una selección. | El sistema verifica que los datos estén correctos, luego muestra el mensaje " Se han guardado los cambios" y posteriormente cuando el usuario exige recomendaciones se muestra el mensaje " No hay recomendaciones para usted. Por favor emita sus preferencias en la página de configuración de su perfil". |
|-------------------------------------------------------|--------------------------------------------------------------------------|-----------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

Tabla 8: Descripción del caso de prueba Gestionar Perfil de usuario

| No | Nombre de campo | Clasificación | Valor Nulo | Descripción |
|----|-----------------|---------------|------------|-------------------------------------------------------------------------------|
| 1 | Preferencias | Select | No | Se debe de seleccionar al menos una categoría para realizar recomendaciones . |

Tabla 9: Descripción de las variables del caso de prueba Gestionar Perfil de usuario

Capítulo 3: Implementación y validación de los recursos del sistema de recomendación para un proveedor de servicio OAI-PMH

| Escenario | Descripción | Cantidad de documentos | Respuesta del sistema | Flujo central |
|--------------------------------|------------------------------------------------------------------------------------------------|-----------------------------|-------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| EC2 Gestionar Clasificador. | El sistema debe realizar automáticamente la clasificación de los documentos de forma correcta. | V 10, 50, 100, Todos. | El sistema clasifica automáticamente los documentos que el usuario administrador o con permisos seleccione. | 1- El usuario administrador desea que el sistema clasifique documentos de forma automática. 2- El sistema brinda la posibilidad de definir cuantos documentos se desean clasificar. 3- Se realiza la clasificación automática basado en los documentos antes clasificados. |

Tabla 10: Descripción del caso de prueba Gestionar Clasificador

| No | Nombre de campo | Clasificación | Valor Nulo | Descripción |
|----|------------------------|---------------|------------|--------------------------------------------------------|
| 1 | Cantidad de documentos | Select | No | El usuario selecciona una de las opciones que aparece. |

Tabla 11: Descripción de las variable del caso de prueba Gestionar Clasificador

| Escenario | Descripción | Título | Respuesta del sistema | Flujo central |
|-------------------------------------------------|---------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| EC1 Desclasificar documento manualmente. | El sistema debe permitirle al usuario administrador desclasificar documentos correctamente. | V Algoritmo para la Generación Automática de Resúmenes de un Documento HTML. | El sistema gestiona la relación entre documentos y preferencias que sirva de base para el proceso de clasificación. | 1- El usuario administrador decide incorporar un documento a una preferencia o desligar un documento de una preferencia. 2- El sistema permite marcar el documento y seleccionar la |
| EC2 Desclasificar documento incorrectamente. | El sistema no debe permitirle al usuario administrador | I El usuario no seleccione ningún documento. | El sistema verifica que los datos estén correctos, si no lo son, muestra | |

Capítulo 3: Implementación y validación de los recursos del sistema de recomendación para un proveedor de servicio OAI-PMH

| | | | | |
|--|---------------------------------------------|------------------------------------|---------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------|
| | desclasificar un documento incorrectamente. | No se selecciona ningún documento. | primeramente el mensaje "Se han guardado las opciones de configuración" y luego "Debe seleccionar al menos un documento". | preferencia o directamente después de marcar el documento da la opción de desclasificar. |
|--|---------------------------------------------|------------------------------------|---------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------|

Tabla 12: Descripción del caso de prueba Gestionar Documentos

| No | Nombre de campo | Clasificación | Valor Nulo | Descripción |
|----|-----------------|---------------|------------|-------------------------------------------------------------------------|
| 1 | Título | Checkbox | No | El usuario puede o no seleccionar un documento para ser desclasificado. |

Tabla 13: Descripción de las variables del caso de prueba Gestionar documentos

| Escenario | Descripción | Categoría | Título | Respuesta del sistema | Flujo central |
|-----------|--------------------------------------------------------------------------------------------------------------|-----------|--------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| EC1 | El sistema debe permitirle al usuario administrador clasificar los documentos manualmente de forma correcta. | V | V | El sistema verifica que los datos insertados son correctos, si lo son, almacena el nuevo Evento en la base de datos. Se muestra el mensaje "Evento <título del evento> se ha creado." | 1- El editor selecciona la opción: "Agregar contenido". 2- Da clic en "Evento" y el sistema muestra una interfaz para ingresar los datos. 3- El editor introduce los datos y selecciona el botón "Guardar". |
| EC2 | El sistema no le | I | NA | El sistema verifica | |

Capítulo 3: Implementación y validación de los recursos del sistema de recomendación para un proveedor de servicio OAI-PMH

| | | | | |
|-----------------------------------------------|----------------------------------------------------|---------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Clasificar documentos manual con datos vacios | permite al editor crear el Evento incorrectamente. | “- Seleccione-” que el usuario no seleccione ninguna de las opciones que brinda el Sistema. | Que el usuario seleccione un documento ejemplo " Tesis de Doctorado " y no seleccione una categoría. | que los datos estén correctos, si no lo son, muestra algunos de los siguientes mensajes: " Se han guardado las opciones de configuración " y seguidamente " Debe seleccionar la categoría a la que pertenecerá el documento". |
|-----------------------------------------------|----------------------------------------------------|---------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

Tabla 14: Descripción del caso de prueba Clasificar documentos manualmente de forma correcta

| No | Nombre de campo | Clasificación | Valor Nulo | Descripción |
|----|-----------------|-----------------|------------|-------------------------------------------------------|
| 1 | Categoría | <i>Select</i> | No | El usuario selecciona una de las opciones que aparece |
| 2 | Título | <i>Checkbox</i> | No | El usuario puedo o no seleccionar un documento |

Tabla 15: Descripción de las variables del caso de prueba Clasificar documentos manualmente de forma correcta

3.4.2 Resultado de las pruebas

Las pruebas efectuadas se realizaron en 2 iteraciones y una revisión final. Dichas pruebas se llevaron a cabo con la ayuda de los diseños de casos de pruebas por cada funcionalidad identificada. En la primera iteración de pruebas de 6 requisitos funcionales implementados, se identificaron 4 no conformidades de las cuales se solucionaron solo 3, en la segunda iteración de 6 requisitos identificados se seleccionaron 5 no conformidades corregidas en su totalidad y en la revisión final se encontraron 0 no conformidades.

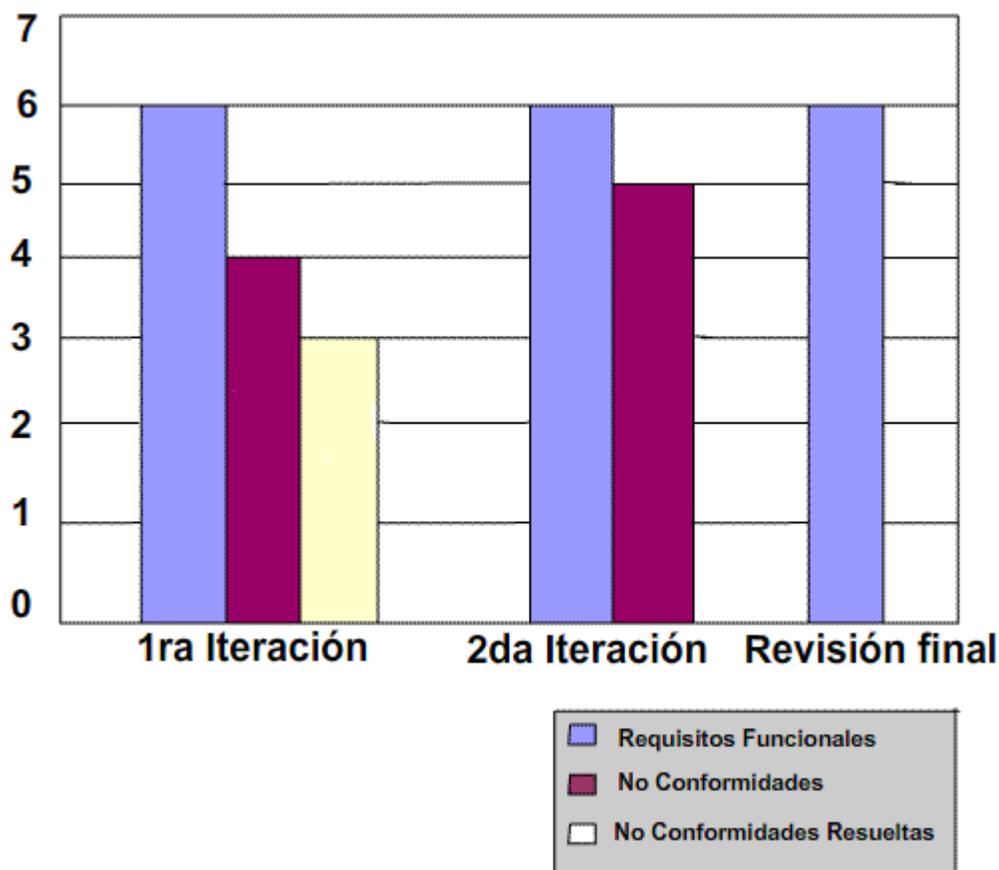


Figura 12: Gráfico de las iteraciones de las pruebas realizadas

3.5 Resultados de las pruebas de carga y estrés

Para llevar a cabo las pruebas de carga y estrés se utilizó la herramienta *Apache JMeter*. Este tipo de prueba se utiliza para simular una carga pesada en un servidor, en la red o a un objeto, para poner a prueba su resistencia o para analizar el rendimiento en diferentes tipos de carga. Para realizar estas pruebas, se simula el comportamiento de varios usuarios conectados al sistema realizando varias transacciones. Para la realización de las pruebas de carga y estrés se efectuó un análisis del tráfico de usuarios en los sistemas homólogos estudiados, lo que arrojó que las visitas a los sistemas de recomendación se encontraban en un rango de 1500 a 4500 visitas mensuales, lo que sería aproximadamente 100 usuarios diarios. Se realizaron las pruebas de carga y estrés en el caso crítico en que se conecten los 100 usuarios concurrentes.

Las pruebas de carga y estrés se desarrollaron con la ayuda de la herramienta *Apache JMeter*, en un ambiente utilizando un ordenador con las siguientes características:

- *Microprocesador Intel DualCore(R) CPU a 2.00GHz.*

Capítulo 3: Implementación y validación de los recursos del sistema de recomendación para un proveedor de servicio OAI-PMH

- Memoria RAM 2GB.

Los resultados obtenidos en las pruebas de carga se consideran satisfactorios debido a que los tiempos de respuestas del servidor se encuentran en el rango de tiempo (de 1 a 6 segundos). La propuesta de solución generó un total de 34253,01 kb/seg de transferencia de datos para 100 usuarios concurrentes esperados, lo que incurrió en un rendimiento de 19,7/ seg. Se demuestra que la propuesta de solución es estable, ya que se mantuvo prestando servicios todo el tiempo sin incurrir en fallos. Además se probó para un total de 20 y 100 usuarios concurrentes, obteniendo los resultados que se muestran en la Figura 13:

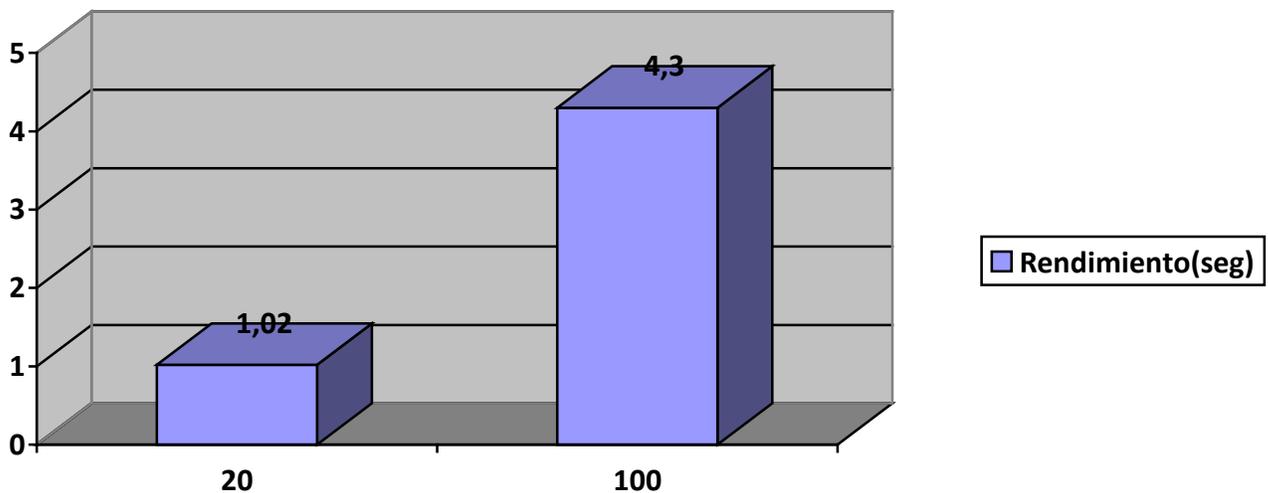


Figura 13: Resultados de las pruebas de carga

Los resultados de las pruebas de estrés se consideran satisfactorios, pues luego de sobrepasar la cantidad de 100 usuarios con un total de 150 y 200 usuarios concurrentes, la propuesta de solución se mantuvo estable prestando servicios todo el tiempo sin incurrir en fallos, arrojando los resultados que se muestran en la Figura 14:

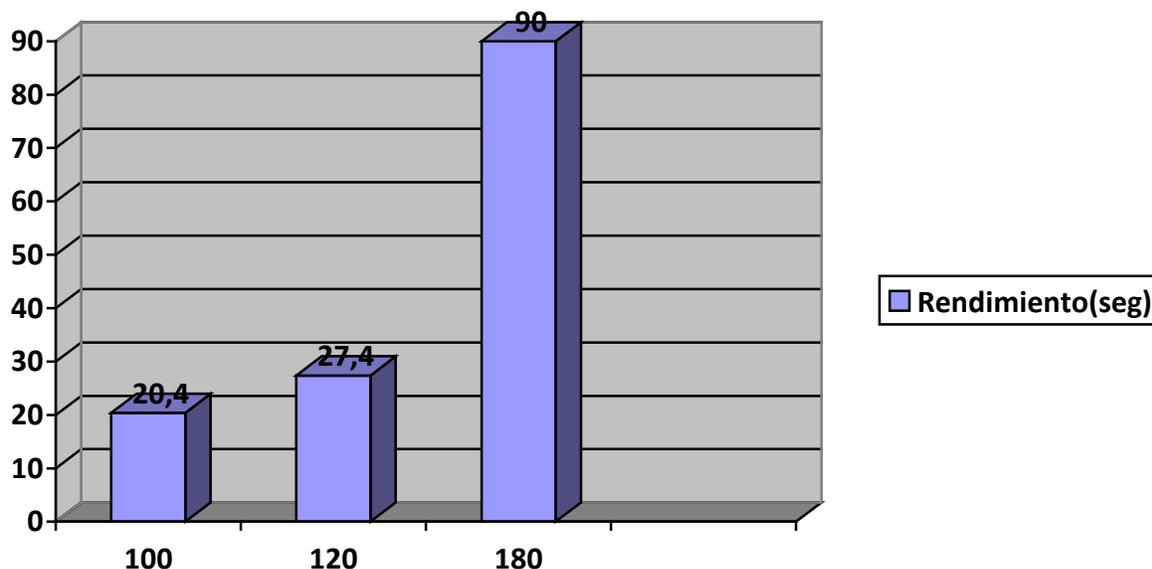


Figura 14: Resultados de las pruebas de carga

3.6 Pruebas de seguridad

Para garantizar la seguridad del sistema se realizan pruebas de seguridad mediante una lista de chequeo aplicable a soluciones *Web*. El objetivo general de esta lista es evaluar a través de indicadores la seguridad de las sistemas en un primer nivel (nivel 1), establecido por los especialistas del grupo de seguridad del Departamento de Pruebas de *Software* (DEPSW) de la UCI. Los especialistas del grupo de Seguridad del DEPSW de la UCI han establecido 3 niveles para realizar las pruebas de Seguridad. Para evaluar la seguridad de las aplicaciones en un primer nivel (nivel 1) se definió una lista de chequeo que cuenta con 15 indicadores separados en 4 tipos de pruebas accesibles a los 3 niveles.

3.6.1 Resultados de las pruebas

El resultado de la evaluación de dicha lista arrojó, para los 4 tipos de pruebas, los siguientes resultados:

- 1. Pruebas de Autorización:** Ningún usuario estándar puede modificar sus privilegios ni los de otro en la aplicación.
- 2. Pruebas de Gestión de Sesiones:** No se puede acceder a la aplicación copiando la URL después de estar autenticado, cerrar el navegador y volver a abrirlo. Así mismo no se puede acceder al cerrar la sesión de un usuario y dar clic en el botón “Atrás” del navegador.

Capítulo 3: Implementación y validación de los recursos del sistema de recomendación para un proveedor de servicio OAI-PMH

3. Validación de Datos: El sistema valida a través del servicio *SoapClient*, que los usuarios pertenezcan al dominio UCI y verifica que la contraseña introducida sea correcta. El sistema no muestra mensajes indebidos al colocar en la barra de dirección o en campos de entrada los caracteres: comillas simples (‘), signos de *ampersand* (&), signos: + - /.

4. Comprobación del Sistema de Autenticación: Los mensajes de error para distintas combinaciones de autenticación muestran la misma información. Los tiempos de respuestas usuario correcto - contraseña incorrecta y usuario - contraseña incorrecta son los mismos. El sistema protege el envío de los datos mediante el protocolo seguro (*HTTPS*).

Esta lista de chequeo permitió recoger los puntos eficientes y los ineficientes que tienen los elementos chequeados, así como verificar que el grado de seguridad de la aplicación es adecuado para proteger la información. Para evaluar la seguridad del sistema se hizo uso de la herramienta *Acunetix Web Vulnerability Scanner* 8.0.

Acunetix Web Vulnerability Scanner: es una herramienta que será capaz de escanear sitios *Web* en busca de posibles fallos de seguridad que puedan poner en peligro la integridad de la página publicada en Internet. Esta aplicación ejecuta una serie de pruebas, totalmente configurables por el usuario, para identificar las vulnerabilidades tanto en la programación de la página como en la configuración del servidor (SCANNERCITADO, 2014). A continuación, se muestran las vulnerabilidades identificadas con el uso de la herramienta:

Durante el escaneo realizado por la herramienta fueron detectadas solo 12 no conformidades, de ellas 8 de prioridad baja, 2 de prioridad media y 2 informativas. Entre las no conformidades de prioridad baja se encontraron: ficheros del servidor Apache vulnerables a ataques de fuerza bruta y *cookies* de sesión sin el indicador de seguridad. Las no conformidades de prioridad media están relacionadas con el auto completamiento de los formularios, se corrigió esta no conformidad. El resto deja al descubierto información acerca de una página de error que contiene la versión del servidor *Web* y una lista de los módulos habilitados en este servidor.

Conclusiones generales

El presente trabajo, ha posibilitado dar solución a un grupo de necesidades del proveedor de servicios OAI-PMH correspondiente a la aplicación DIMA. Específicamente aquellas relacionadas con la necesidad de existencia de un sistema de recomendación para facilitar la búsqueda de los usuarios. Las principales conclusiones a las que se llega después de llevar a cabo la investigación y el desarrollo de la aplicación son:

- ❖ La tipología de sistemas de recomendación basados en contenido es el más adecuado para un entorno OAI-PMH, fundamentalmente por la naturaleza de los datos que se manejan.
- ❖ La metodología *OpenUP* se adapta al proceso completo del sistema por los artefactos que propone y por su flexibilidad.
- ❖ La arquitectura modular del sistema de gestión de contenidos *Drupal* y el concepto de entidad dotan a las aplicaciones desarrolladas sobre esta plataforma la posibilidad de extenderse y adaptarse con relativa facilidad.
- ❖ Los metadatos título y palabras claves correspondiente al estándar *Dublin Core* aportan un conjunto de palabras muy precisas y por ende son los mejores metadatos para emplearlos en la comparación de documentos. De esta forma tributan de forma significativa al proceso de recomendación.
- ❖ La representación de los documentos como vectores y el uso de la medida de similitud del coseno, se adaptan perfectamente al escenario de la documentación académica y científica por la facilidad de encontrar los valores de cada componente del vector.
- ❖ La recomendación como proceso automático complementa la función fundamental de los proveedores de servicio OAI-PMH y le facilitan al usuario final la tarea de la recuperación de información.

Recomendaciones

Luego de haber realizado el desarrollo del sistema de recomendación para un proveedor de servicios OAI-PMH y a su vez haberse cumplido cabalmente con los objetivos trazados se exponen a continuación las siguientes recomendaciones:

- ❖ Continuar desarrollando nuevas funcionalidades que aunque no sean críticas, hagan a la solución más competitiva, eficiente y fácil de usar por el usuario.
- ❖ Continuar el estudio de los sistemas de recomendación con motivo de mejorar su entendimiento y lograr mejores propuestas de automatización.
- ❖ Pedir diferentes tipos de criterios a los usuarios acerca de sus recomendaciones para de esta forma tener un mayor acercamiento a sus preferencias.
- ❖ Continuar el seguimiento de las actualizaciones de las herramientas y tecnologías usadas para garantizar mejoras en versiones futuras de la solución, además de extender la búsqueda de nuevas tecnologías que en lo posible puedan adoptarse y perfeccionar la solución.

Bibliografía y Referencias bibliográficas

1. ABREU, B., YANEDI. DOMINGUEZ, CRUZ, LUIS. FERNÁNDEZ, FERNÁNDEZ, MAIKEL MANUEL, LA HABANA, CUBA, CUBA, (2011). La biblioteca digital personalizable como complemento del sistema de bibliotecas de las universidades
2011, vol. 22, nº Disponible en: <http://acimed.sld.cu/index.php/acimed/article/view/150/140>
2. ALDERSON, C. J., CLAPHAM, C. Y WALL, D. (1995). *Exámenes de idiomas, elaboración y evaluación* Madrid: Cambridge University Press: Disponible en:
http://cvc.cervantes.es/ensenanza/biblioteca_ele/diccio_ele/diccionario/item.htm
3. ALEGSA.ALEGSA.COM.AR. (2010). *Definición de IDE de 2014*. Disponible en:
<http://www.alegsa.com.ar/Dic/ide.php>.)
4. ANTUNES, N. L., E (2009). *Effective detection of SQL/XPath injection vulnerabilities in web services* Disponible en: <http://www.acunetix.com/vulnerability>
5. BARRANCO, M. J. L. P., G. (2006). *Un Sistema de Recomendación Basado en Conocimiento con Información Lingüística Multigranular de 2014*. Disponible en:
http://sinbad2.ujaen.es/cod/archivosPublicos/publicaciones/congresos/Martinez2006b_SIGEF.pdf
6. BELLO, L. (2012). *Portal web para la biblioteca de la Universidad de las Ciencias Informáticas*. Presentación en evento, Universidad de las Ciencias Informáticas, 2012.
7. BORDIGNON, F. T., G. (2007). *Recuperación de información: un área de investigación en crecimiento*. 2007, vol. Vol 6,
8. BROWN, A. A., YUNIER (2012). *Desarrollo de un Sistema de Recomendación Basado en Contenido para la Plataforma Video*. [científico]. 2012, Disponible en:
http://repositorio_institucional.uci.cu/jspui/handle/ident/2
9. CAMPOS, D., D. VÁZQUEZ, ACOSTA, M. (2007). *Implementacion de algoritmos de agrupamiento*. Universidad de las Ciencias informáticas, 2007.
10. EGUÍLUZ PÉREZ, J. (2011). *Introducción a CSS* Disponible en: <http://librosweb.es/css/>
11. FEBLES ESTRADA, A. C. G., TAYCHÉ ; LEÓN PERDOMO, (2011). Una experiencia novedosa para el *testing* desarrollada por un departamento de pruebas de software. 2011, nº Disponible en:
<http://rcci.uci.cu/index.php/rcci/article/view/84/72>
12. FERIA, L. (2012). *Desarrollo de un sistema de recomendación colaborativo para la plataforma VideoWeb*. Investigativa, Universidad de las Ciencias Informáticas, 2012.
13. FERNÁNDEZ, M. y DOMÍNGUEZ, L. (2014). *Sistema para la recuperación de información académica y científica*. Forum de Ciencia y Técnica 2014, Universidad de las Ciencias Informática, 2014.
14. FERNÁNDEZ, M. P. E. U. (2012). *El protocolo OAI-PMH, componente tecnológico para el acceso abierto*. [Documento]. La habana: 2012, vol. ,
15. FERRERA, Y. (2011). *Sistema de recomendación basado en las preferencias de los usuarios para la plataforma PTARTV*. Universidad de las Ciencias Informáticas, 2011.

16. FONT, M. S. (2009). *Sistemas de recomendación para webs de información sobre la salud*. 2009.
17. GARCÍA, E. (2012). *Herramienta JMeter Apache 4.0 de 2014*]. Disponible en: <http://scrum-qa.blogspot.com/2010/09/pruebas-de-stressjmeter.html/>
18. GONZÁLEZ, R., ANTONIO. OSORIA, ALMANZA, FRANK. PABLO. (2011). *Migración de los Servicios Básicos del Portal D'TIC, Centro Virtual de Recursos a Drupal 6*. Universidad de las Ciencias Informaticas, 2011.
19. HERNÁNDEZ, D., ALICIA. SANTANA, FERNÁNDEZ RAMÓN (2013). *Componente de clasificación de huellas dactilares de 2014*]. Disponible en: https://netbeans.org/community/releases/61/index_es.html.
20. HERRERA, E. P., CARLOS HIDALGO, LORENZO. (2004). *Sistemas de recomendaciones de 2014*]. Disponible en: <http://www.hipertext.net>
21. HIDALGO, A. R. V., JOSUÉ SORÍ (2008). *MODELACIÓN DE ARQUITECTURA PARA APLICACIONES EMPRESARIALES EN PHP*
Disponible en: http://repositorio_institucional.uci.cu/jspui/bitstream/ident/TD_1500_08/1/TD_1500_08.pdf
22. HUITRÓN, A. S. (1997). *Creación de grupos virtuales en bibliotecas digitales de 2014*]. Disponible en: <http://ict.udlap.mx/people/lulu/tentativa.html>
23. JACOBSON, I. B., GRADY RUMBAUGH, JAMES (2000). *El proceso unificado de desarrollo de software*. Madrid: Pearson Educacion,
24. KONSTAN, J. T., L. RIEDL, J. HERLOCKER (2004). *Evaluating Collaborative Filtering Recommender Systems de 2014*].
25. LARMAN, C. (1999). *UML y Patrones*. 1999
26. MADDISON, R. N. (1983). *Information System methodologies de 2014*].
27. MAS-COLELL, W., GREEN (1995). *Microeconomic Theory*. New York: Oxford University Press, 1995.
28. MOLINA, G., JESÚS. (2005). *Análisis y Diseño del Software*. 2005, nº
29. ORÉ B., I. A. (2009). *Pruebas funcionales*. 2009, nº Disponible en: http://www.calidadyssoftware.com/testing/pruebas_funcionales.php
30. PIATTINI, M. (1996). *Análisis y Diseño Detallado de Aplicaciones Informáticas de Gestión*
31. PRESSMAN, R. S. (2007). *Ingeniería del Software*. 2007. 2007.
32. REYNOSO, C. K., NICOLÁS. (2004). *Estilos y Patrones en la Estrategia de Arquitectura de Microsoft* Universidad de Buenos Aires:
33. RÍOS SALGADO, S. R. (2013). *Desarrollo de un sistema de difusión de gestión del conocimiento de la ESPE, aplicando la metodología OPENUP Y EL FRAMEWORK RUBY ON RAILS SANGOLQUÍ / ESPE / 2013*. ed. ARTÍCULO CIENTÍFICO. Disponible en: <http://repositorio.espe.edu.ec/handle/21000/6316>

34. RODRÍGUEZ (2014). *Curso de creación y gestión de portales Web con Drupal 7* La habana. Cuba: Disponible en: http://drupaleros.uci.cu/system/files/drupal_7_-_nivel_avanzado.pdf.
35. RODRÍGUEZ, J. A. (2004). *La estructura de los documentos en el ámbito de recuperación de información: propuestas para su compresión, indexación y recuperación* Tesis doctoral, Universidad de Valladolid, 2004.
36. RODRÍGUEZ, M. O., MEYLIN. (2007). *LA METODOLOGÍA XP APLICABLE AL DESARROLLO DEL SOFTWARE EDUCATIVO EN CUBA*. Facultad 9. UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS, 2007.
37. S.A., S. S. A.-S. (2013). *Técnicas y herramientas de tecnología orientada a objetos*: Mendoza - Argentina Disponible en: <http://www.sparxsystems.com.ar/about.html>
38. SANTANA, A. P., YANDRY. (2013). *Sistema de recomendación de objetos de aprendizaje en el Repositorio de Objetos de Aprendizaje RHODA*. Investigativa, Universidad de las Ciencias Informáticas, 2013.
39. SCANNERCIDADO, A. W. V. (2014). *Marco de desarrollo de la Junta de Andalucía*. Disponible en: <http://www.juntadeandalucia.es/servicios/madeja/contenido/recurso/656>
40. SECURITY.PHP, C. P. (2010). *Security Guide* de 2014]. Disponible en: <http://phpsec.org/projects/guide>
41. SERNA M., E. (2011). Análisis y comparación de las propuestas recientes para diseñar casos de pruebas desde casos de usos orientados a verificar los aspectos del funcionamiento del software. 2011, nº Disponible en: http://www.bdigital.unal.edu.co/6114/1/71617999._2012.pdf
42. SOSA, V., LILIBET . ALVAREZ, FORTE, MEDARDO. (2009). *TeleIdentificador Personal. Portal Web*. Tesis de diploma, Universidad de las Ciencias Informáticas, 2009.
43. SUÁREZ, A. F., FERNÁNDEZ, MAIKEL MANUEL DOMÍNGUEZ, LUIS (2012). *Versión 2.0 del proveedor de servicios OAI-PMH para el Sistema de Gestión de Contenidos Drupal*. Investigativa, Universidad de las Ciencias Informáticas, 2012.
44. SUÁREZ ROMERO, A. F. F., M. M. D. CRUZ, L. (2012). *Versión 2,0 del proveedor de servicios OAI-PMH para el Sistema de Gestión de Contenidos Drupal*. Universidad de las Ciencias Informáticas, 2012.
45. TERRY, B. L., D. (2002). *Introducción a Herramientas CASE y System Architect* Disponible en: <http://www.pol.una.py/archivos/IngeInfo/ingeSoftI/MaterialPrimeraC.pdf>
46. TOLOSA, G. H. F. R. A. (2011). *Introducción a la Recuperación de Información: Conceptos, modelos y algoritmos básicos*.
Universidad Nacional de Luján, Argentina:
47. VALDÉS, Y. C., LIZANDRA (mayo 2010). *Sistema de búsqueda de términos y definiciones estandarizadas sobre la Ingeniería de Software (SEVOCATEIS)*. Universidad de las Ciencias Informáticas, mayo 2010.

48. ZAMORA, Y. T. (2013). *Sistema de Recomendación, para el catálogo en Línea de la biblioteca de la Universidad de las Ciencias Informáticas*. Investigativa, Universidad de las Ciencias Informáticas, 2013.

Glosario de término

- ❖ **CSS:** Hojas de Estilos en Cascada (CSS del inglés *Cascading Style Sheets*).
- ❖ **SP:** Proveedor de servicios (SP del inglés *service provider*).
- ❖ **CMS:** Sistema de Gestión de Contenidos (CMS del inglés *Content Management Systems*).
- ❖ **BSD:** Distribución de *Software Berkeley* (BSD del inglés *Berkeley Software Distribution*).
- ❖ **DP:** Proveedor de datos (DP del inglés *data provider*).
- ❖ **HTTP:** Protocolo de Transferencia de Hipertexto (HTTP del inglés *Hypertext Transfer Protocol*).
- ❖ **OAI-PMH:** Iniciativa de Archivos Abiertos - Protocolo para la Transmisión de Metadatos (OAI-PMH del inglés *Open Archives Initiative – Protocol for Metadata Harvesting*).
- ❖ **PHP:** Procesador de Hipertextos (PHP del inglés *Hypertext Pre-processor*).
- ❖ **RUP:** Proceso Unificado de Rational (RUP del inglés *Rational Unified Process*).
- ❖ **SI:** Sistemas de información.
- ❖ **SP:** Proveedor de servicios (SP del inglés *service provider*).
- ❖ **TCP/IP:** Protocolo de Control de Transmisión / Protocolo de *Internet*(TCP/IP del inglés *Transmission Control Protocol / Internet Protocol*).
- ❖ **W3C:** Consorcio de la *Web* o Telaraña Mundial (W3C del inglés *World Wide Web Consortium*).
- ❖ **XHTML:** Lenguaje de Marcado de Hipertexto Extensible (XHTML del inglés *eXtensible Hypertext Markup Language*).
- ❖ **XML:** Lenguaje de Marcado Extensible (XML del inglés *Extensible Markup Language*).

Anexos

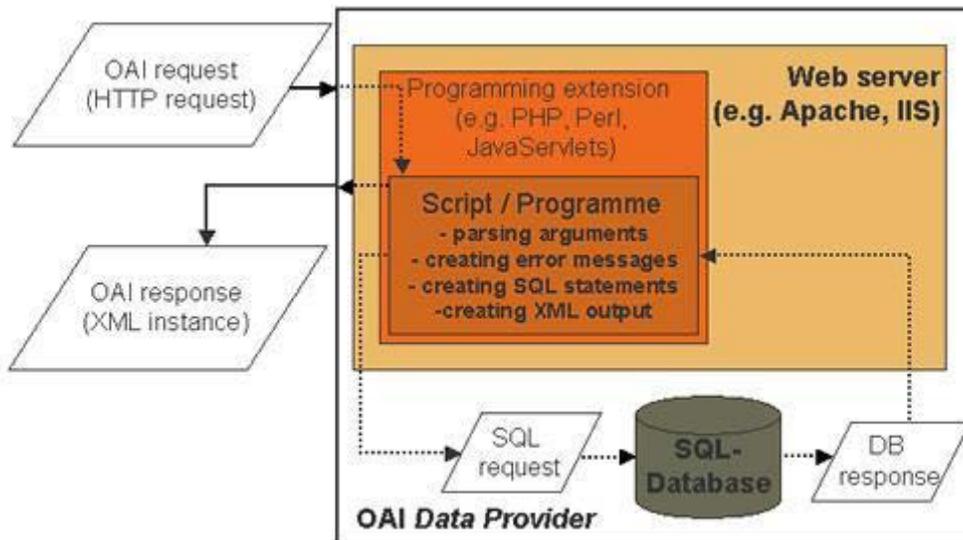


Figura 15: Describe la estructura y el flujo de trabajo

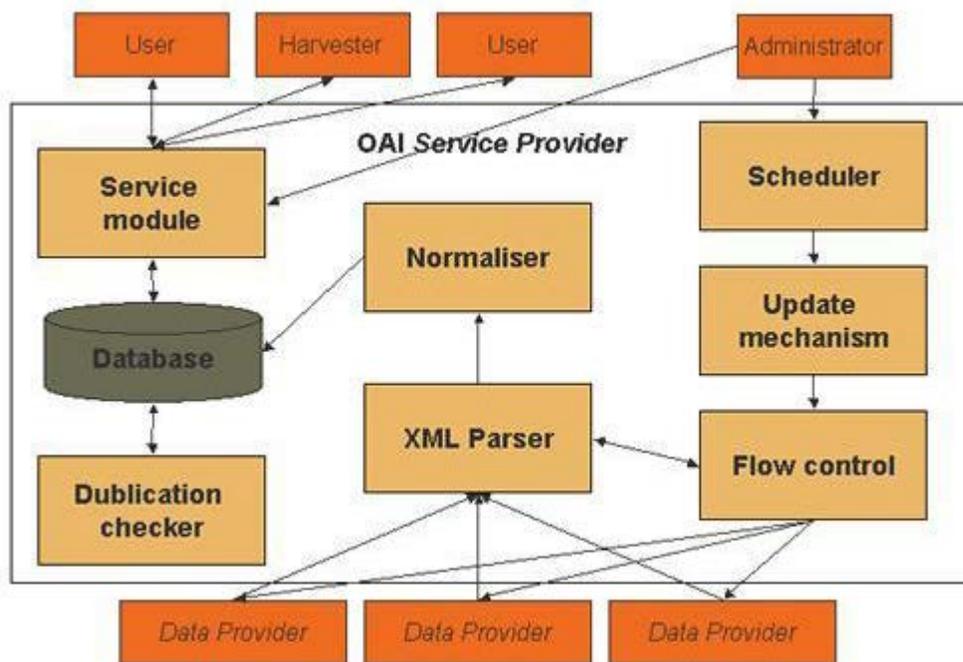


Figura 16: Iteración de los componentes del proveedor de servicios

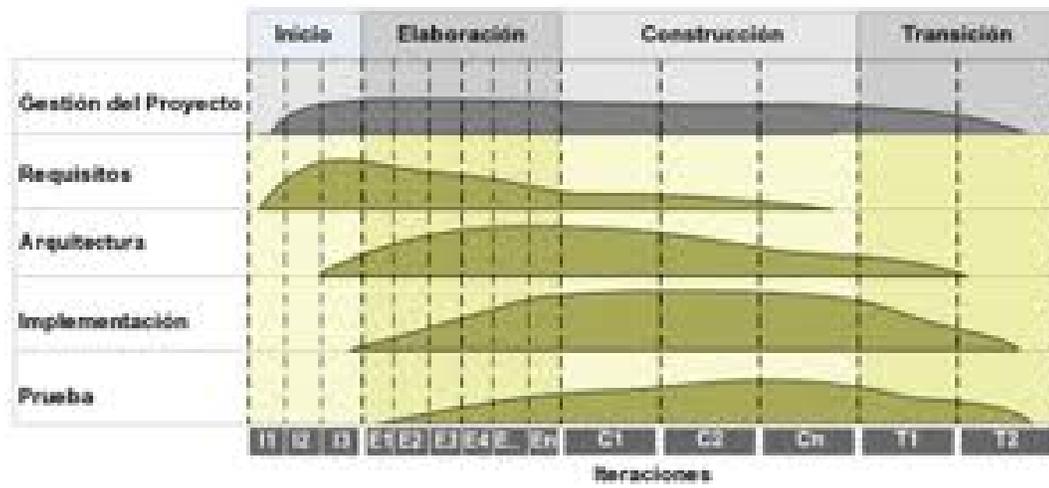


Figura 17: Etapas e iteraciones de la metodología *OpenUP*