Universidad de las Ciencias Informáticas

Facultad 2

Módulo Recomendaciones del sistema para repositorios digitales REPXOS 3.0

Trabajo de Diploma para optar por el Título de

Ingeniero en Ciencias Informáticas

Autores: Bexi Ruiz Ricardo

Cristina de la Caridad Vaillant Valdéz

Tutor(s): Ing. Gleidis Yuriannis Rosabal Espinosa

Ing. Yorjander Tan Guevara

La Habana, Junio de 2015

"Año 57 de la Revolución"

"Estoy convencido que lo que separa a los emprendedores exitosos y los que no es la ***perseverancia****. Es tan difícil y pones tanto de tu vida en esto, hay momentos tan duros en que la mayoría se da por vencido, no los culpo, es muy difícil y consume gran parte de tu vida. A menos que tengas mucha* ***pasión*** *en lo que haces no vas a sobrevivir, vas a darte por vencido"*

Steve Jobs.

DECLARACIÓN DE AUTORÍA

Declaramos que somos los únicos autores de este trabajo y autorizamos a la Universidad de las Ciencias Informáticas a hacer uso del mismo en su beneficio.

Para que así conste firmamos la presente a los 18 días del mes Junio del año 2015.

Autores:

_____	_____
Bexi Ruiz Ricardo	Cristina de la Caridad Vaillant Valdéz

Tutores:

_____	_____
Ing. Gleidis Yuriannis Rosabal Espinosa	Ing. Yorjander Tan Guevara

DATOS DE CONTACTO

Tutores:

Ing. Gleidis Yuriannis Rosabal Espinosa: Graduada en el año 2011 como Ingeniera en Ciencias Informáticas, en la Universidad de las Ciencias Informáticas. Se desempeña como especialista B en ciencias Informáticas en el Departamento de Aplicaciones del Centro CIGED ocupando el rol de analista del proyecto Repositorio Institucional. Correo electrónico: gyrosabal@uci.cu.

Ing. Yorjander Tan Guevara: Graduado en el año 2013 como Ingeniero en Ciencias Informáticas, en la Universidad de las Ciencias Informáticas. Se desempeña ocupando el rol de desarrollador en el proyecto Repositorio Institucional del Centro de Informatización de la Gestión Documental. Correo electrónico: ytan@uci.cu

DEDICATORIA

Dedico mi tesis a toda mi familia en especial a mi mamá Isabel.

A mi hermano Migue, a mi papa, a mis tíos Inés, Ramiro y Yudit.

A mi abuela Trina por brindarme todo el amor del mundo y por estar ahí siempre para mí.

A mis primos Ivett, Ronny, Camila y Ashley.

A mi novio Angel: gracias por todo mi vida TE AMO.

A mi compañera de tesis Cristina.

A mis amigos más cercanos Rodrigo, Adriana, Claudia, Yuliet, Jeffrey, Luisma, Aramis y Amigo.

A mi abuelito "Nando" que ya no está junto a mí pero sé que estaría orgulloso de que haya logrado mi sueño.

Bexi

A mi madre y mi padre por ser todo en mi vida, lo que más yo quiero y brindarme su apoyo incondicional.

A mi novio Luis Manuel por acompañarme los 5 años de mi carrera universitaria, ser mi amigo fiel, quererme y apoyarme en todo momento.

A mi queridísima compañera de tesis Bexi por aguantar todos mis caprichos.

A mí misma por ser luchadora, dedicada y llegar hasta aquí.

Cristina

AGRADECIMIENTOS

Primero que todo quiero agradecer a mi compañera de tesis por su esfuerzo, dedicación y comprensión durante todo este tiempo de trabajo y sacrificio, esto fue lo que hizo posible el resultado de esta investigación.

Gracias a mis queridos formadores en especial a los profes Arelis Rivero, Cesar Richard, Miguel Angel Marín y Alejandro Castellini.

Gracias de corazón a mi tutora Gleidis y a mi arquitecto de proyecto Luis Carlos por todo el apoyo que nos brindaron y que siempre estuvieron ahí cuando los necesitamos.

Gracias a los buenos amigos y compañeros que conocí durante todo este tiempo en especial a Rodrigo, Claudia, Yuliet, Aramis, Jeffrey, Amigo, Deylert y Luisma.

Gracias a mi novio por su amor, dedicación y comprensión en los momentos más difíciles y por estar siempre ahí para mí.

Gracias a mi familia mi más grande tesoro en especial a mi hermano Migue, a mis tíos Inés, Yudit y Ramiro, a mi papa, a mis primos y a mi cunada Claudia.

GRACIAS a mi madre por todo su amor, cariño y confianza. GRACIAS por darme las fuerzas y los consejos necesarios para salir adelante. Este logro es tan tuyo como mío, mil GRACIAS y TE AMO.

Bexi

AGRADECIMIENTOS

A mi compañera de tesis Bexi, y más que eso amiga por estar a mi lado y dedicar tanto amor y esfuerzo para que fuera posible la realización de esta tesis.

A mis padres Nilda y Jorge Luis por apoyarme, aconsejarme y guiarme siempre hacia lo mejor.

A mi novio Luis Manuel por ser un amigo fiel, brindarme su apoyo y quererme.

A mi tutora Gleidis que a pesar de fajarnos la quiero mucho y aprecio su apoyo y dedicación.

A mis profesores Arelys y Ángel Fabra por apoyarme y aconsejarme en todo momento.

A Luis Carlos y Deyler por ayudarnos y ser atentos ante cualquier duda.

A mis amigos Karla, Annia, Jeffrey Díaz y Mario (el más fashion) por apoyarme desde que los conocí, comprenderme y brindarme su apoyo siempre.

A mis compañeros Dariel, Yanelis, Adyana y Yuliet por estar a mi lado.

A mis compañeros de grupo y de futbol, en especial a Yeni.

Cristina

RESUMEN

Los sistemas de recomendación son utilizados mundialmente en diversos campos entre los que se destaca la investigación científica. Actualmente el proyecto Repositorio Institucional, del Centro de Informatización de la gestión documental (CIGED) de la Universidad de las Ciencias Informáticas (UCI) desarrolla el sistema REPXOS 3.0. Este sistema es el encargado de gestionar todas las producciones científicas e investigativas generadas en la universidad. El cúmulo de información existente en REPXOS 3.0 provoca que los usuarios al realizar las búsquedas no siempre obtengan como resultado en primera posición los ítems más relevantes de acuerdo a sus preferencias y se dificulte la localización de dichos ítems. Para solucionar este problema, se concibió el desarrollo del Módulo Recomendaciones que contribuyó a la recuperación de los ítems más relevantes para sugerirlos a los usuarios del sistema. Para ello se realizó un estudio y análisis de los sistemas de recomendación existentes. Como resultado de los estudios realizados, se definió el Módulo Recomendaciones como un sistema de recomendación basado en el filtrado colaborativo con un enfoque basado en memoria y de este el esquema Usuario-Usuario. Se obtuvo como resultado un módulo para el sistema REPXOS 3.0 que realiza recomendaciones automáticas y manuales de ítems a usuarios. El mismo brinda la posibilidad de compartir ítems entre usuarios del sistema que se consideren interesantes para el desarrollo de investigaciones científicas.

Palabras claves: enfoque basado en memoria, esquema usuario-usuario, filtrado colaborativo, ítem, sistemas de recomendación.

TABLA DE CONTENIDOS

INTRODUCCIÓN

El avance científico tecnológico y el desarrollo de las nuevas tecnologías de la información y las comunicaciones (TIC), ha propiciado que la red Internet se convierta en un fenómeno social que ha evolucionado con el devenir de los años. En la actualidad, gracias a Internet se ha puesto al servicio de los usuarios, gran cantidad de información que abarca diversos contenidos. En ocasiones esta información no se encuentra organizada y debidamente estructurada, lo cual trae como consecuencia que se emplee mucho tiempo para analizarla, clasificarla y seleccionar el contenido que sea de mayor interés, según el criterio de búsqueda que se haya especificado. “La gran cantidad de sitios especializados en Internet, ofertando millones de productos y/o servicios para su consumo, se convierte en un caos de información sin solución cuando se necesita realizar una adquisición eligiendo entre todas las opciones existentes” (1).

Para facilitar la búsqueda de información o productos acorde a las necesidades de los usuarios surgen los Sistemas de Recomendación (SR). “Un Sistema de Recomendación tiene como principal tarea seleccionar ciertos objetos de acuerdo a los requerimientos del usuario” (2). Estos sistemas son capaces de reducir la sobrecarga de información al brindar los resultados de búsqueda que más se ajustan a las necesidades de los usuarios y proporcionar acceso a la información personalizada para un dominio en concreto.

Los SR proveen como salida una serie de productos, servicios o contenidos, que se ajustan a las necesidades de los usuarios. Con la información que se tiene de los usuarios se pueden conocer sus gustos o preferencias y de esta forma describir qué productos le han interesado en el pasado y en qué medida. Esta información se puede obtener implícitamente, explícitamente o mediante la combinación de ambas. Los SR funcionan en correspondencia con actitudes que existen desde que el ser humano tiene conciencia e inteligencia:

- Pedir consejo o recomendación a expertos en la materia o seguir a aquellos individuos que tienen gustos similares al del usuario.
- Seleccionar objetos que tienen características similares a objetos que le hayan gustado anteriormente o que se parecen al que inicialmente buscaba.

Estas dos tendencias han dado lugar a las dos ramas principales de los SR: los colaborativos y los basados en contenido (1). Los primeros se basan en la similitud de gustos entre usuarios y los segundos en la similitud entre objetos. De acuerdo a su funcionamiento los SR también se clasifican en: basados en conocimiento, en utilidad, demográficos e híbridos. Para generar las recomendaciones los SR emplean diferentes algoritmos

que permiten realizar el filtrado de la información. Uno de los algoritmos de filtrado más utilizados es: k vecinos más cercanos (3).

La selección del algoritmo a emplear se basa en buscar el que mejor se ajusta al conjunto de datos que va a ser manejado por el SR. El algoritmo de recomendación es el corazón del SR y permite asociar criterios de búsqueda con valoraciones de los productos para filtrar la información y devolver resultados más relevantes. El manejo de los SR permite que las búsquedas sean más efectivas para los usuarios ya que estos alcanzan una mayor satisfacción al obtener los resultados que más se ajustan a sus preferencias. El desarrollo de los SR ha sido ampliamente estudiado, estos son utilizados en diversos campos como el entretenimiento (4), la investigación científica (4), el comercio (5), la industria (4), entre otros. El empleo de los SR en el campo de la investigación científica contribuye al avance tecnológico, mejora la calidad de las producciones científicas e investigativas y favorece al desarrollo y la innovación; elementos que facilitan la informatización de la sociedad.

Como parte del proceso de informatización de la sociedad cubana surge la Universidad de las Ciencias Informáticas (UCI). Esta está constituida por diferentes centros de producción, uno de ellos es el Centro de Informatización de la Gestión Documental (CIGED). Este centro está compuesto por diferentes proyectos, entre ellos el proyecto Repositorio Institucional que actualmente desarrolla el sistema REPXOS 3.0. Este sistema gestiona todas las producciones científicas e investigativas generadas en la universidad con el objetivo de mantenerlas centralizadas y disponibles. Además permite el almacenamiento, la visualización y la obtención de ítems (documento digital con sus metadatos[1] asociados) mediante búsquedas simples y avanzadas. A pesar de las ventajas que ofrece, este todavía presenta un conjunto de deficiencias que serán descritas a continuación:

- Debido al cúmulo de información existente en el sistema, los resultados de búsqueda no siempre muestran en primera posición los ítems más relevantes de acuerdo a las preferencias de los usuarios, lo cual hace difícil la localización de dichos ítems.
- Esto provoca que aumente el tiempo de adquisición de la información para los usuarios.

[1] Conjunto de atributos o elementos necesarios para describir un recurso determinado, que funciona como identificador de los materiales digitales diseñados.

- Los usuarios no pueden socializar con otros usuarios, a través del sistema, información relevante que consideran que es de su mismo interés.

De la situación anterior se propone como **problema a resolver**: ¿Cómo contribuir a la recuperación de los ítems más relevantes para los usuarios del sistema REPXOS 3.0 según sus preferencias?

Se define como **objeto de estudio**: los sistemas de recomendación, siendo el **campo de acción**: sistemas de recomendación de filtrado colaborativo.

Para la solución del problema se plantea como **objetivo general**: Desarrollar un Módulo Recomendaciones basado en el filtrado colaborativo para el sistema REPXOS 3.0 que contribuya a la recuperación de los ítems más relevantes para los usuarios según sus preferencias.

Del **objetivo general** se definen los siguientes **objetivos específicos**:

- Caracterizar las tendencias actuales de los sistemas de recomendación basados en el filtrado colaborativo en el ámbito nacional e internacional.
- Desarrollar el Módulo Recomendaciones del Sistema para Repositorios Digitales REPXOS 3.0 a partir de las funcionalidades definidas.
- Aplicar pruebas para validar la solución.

El presente trabajo brindará el siguiente aporte a los usuarios del sistema REPXOS 3.0:

- Permitirá que usuarios con experiencia realicen recomendaciones a otros usuarios creando así un área de intercambio de información.
- Permitirá que los usuarios evalúen los ítems consultados de acuerdo a su nivel de aceptación sobre los mismos.
- Realizará recomendaciones automáticas y manuales teniendo en cuenta las necesidades e intereses de los usuarios.

Para realizar esta investigación se utilizaron los siguientes **métodos científicos**:

Teóricos:

- **Histórico Lógico:** se empleó este método para analizar la evolución histórica de los sistemas de recomendación.

- **Analítico – Sintético**: permitió analizar las características de los sistemas de recomendación existentes y definir cuál será utilizado en el Módulo Recomendaciones.
- **Modelación**: se utilizó para modelar los diferentes artefactos propuestos por cada fase de la metodología de desarrollo empleada por el Módulo Recomendaciones.

Empíricos:

Entrevista: se realiza una entrevista a los especialistas del proyecto Repositorio Institucional, mediante la cual se obtiene toda la información posible para el desarrollo del Módulo Recomendaciones para el sistema REPXOS 3.0, procesos de negocio, requisitos funcionales y no funcionales.

CAPÍTULO 1. **FUNDAMENTACIÓN TEÓRICA PARA EL DESARROLLO DEL MÓDULO RECOMENDACIONES DEL SISTEMA PARA REPOSITORIOS DIGITALES REPXOS 3.0**

En este capítulo se realiza un estudio de los principales conceptos asociados a los sistemas de recomendación. Se exponen los antecedentes de los sistemas de recomendación actuales y las características que permiten comprender su funcionamiento. Además se realiza un estudio de las tendencias actuales en el desarrollo de sistemas de recomendación. Se describen brevemente la metodología, las herramientas, tecnologías y el lenguaje, que se utilizan en el desarrollo del módulo.

1.1. Conceptos asociados

A continuación se definirán algunos conceptos relacionados con los sistemas de recomendación enfocados al campo de la investigación científica.

Repositorio Digital

Los repositorios, también conocidos como repositorios digitales, están constituidos por un conjunto de archivos digitales en representación de productos científicos y académicos que pueden ser accedidos por los usuarios. Específicamente, los repositorios institucionales consisten en estructuras web interoperables de servicios informáticos, dedicadas a difundir la permanencia de los recursos científicos y académicos (físicos o digitales) de las universidades a partir de la enumeración de un conjunto de datos específicos (metadatos), para que esos recursos se puedan recopilar, catalogar, acceder, gestionar, difundir y preservar de forma libre y gratuita.(6)

Documento Digital

Documento: cualquier unidad significativa de información que haya sido registrada en un soporte que permita su almacenamiento y su posterior recuperación. Ejemplo de documento lo constituyen: los libros, las revistas y los artículos que en ellas se publican; las cartas, las facturas, y los informes escritos de diverso tipo. Pero son documentos también los dibujos, los gráficos, las fotografías, las películas, los videos y las distintas formas de sonido grabado como discos, cintas, cartuchos o "*cassetes*". Son documentos -unidades significativas de información registrada- sin importar su forma mediática: texto, imagen fija, imagen en movimiento o sonido; sin importar el

soporte en que se registra la información: papel, cuero, cortezas, celulosa, acetato o cinta magnética. Son documentos impresos unos y audiovisuales, los otros. (7)

Documento Digital: un documento que no recibe su nombre de la forma mediática en que se presenta la información, sino directamente de la manera particular en que se registra. En estos la información, no importa su forma: texto, sonido, imagen fija o en movimiento, se registra en un medio. (7)

Ítem

En el sistema REPXOS 3.0 un ítem lo constituye la unión de un documento digital con sus metadatos asociados y el formato de metadato que se emplea es el *Dublin Core* (DC). Este es un formato de metadatos definido por consenso internacional. El Conjunto de Elementos de Metadatos *Dublin Core* (DCMES, por sus siglas en inglés) define dos niveles simple y cualificado. El *Dublin Core* Simple presenta quince elementos para la descripción y búsqueda de recursos sencillos. Estos elementos son: título, creador, materia, descripción, editor, colaborador, fecha, tipo de recurso, formato, identificador, fuente, idioma, relación, cobertura, derechos. El *Dublin Core* Cualificado presenta otros elementos adicionales, como la Audiencia, Procedencia, Titulares de los derechos, etc. (8)

1.2. Sistema de Recomendación

Con el desarrollo tecnológico que se ha producido en los últimos años se ha extendido el uso del internet, esto ha provocado un crecimiento en el número de sitios web, documentos digitales, entre otros generando una gran sobrecarga de información (1). Debido a esto se hace cada vez más complejo para el usuario encontrar la información que realmente le interesa. Para darle solución a este problema surgen los Sistemas de Recomendación (SR), los cuales producen recomendaciones personalizadas como salida además de guiar al usuario hacia productos interesantes o útiles entre una gran cantidad de productos disponibles (9). De esta forma el usuario empleará menos tiempo en encontrar lo que necesita de manera rápida y cómoda (1).

1.2.1. Antecedentes de los Sistemas de Recomendación

A principios de la década de los 90 empezaron a surgir dentro de los servicios de grupos de noticias, servicios de filtrado de noticias, que permitían a su comunidad de usuarios acceder exclusivamente a aquellas noticias que potencialmente podían ser de su interés (10). No

obstante, el primer sistema de recomendación que apareció fue el llamado "*Tapestry*" (2), desarrollado por XeroX PARC (2). Tapestry permitía a sus usuarios encontrar documentos basados en comentarios hechos previamente por otros usuarios. Al ser un experimento pionero surgieron muchos problemas ya que solo funcionaba correctamente con pequeños grupos de personas y eran necesarias consultas de palabras específicas para obtener resultados. También tenía otras carencias como la falta de privacidad. A pesar de todo fue un sistema que resultó crucial para el posterior desarrollo de los sistemas de recomendación colaborativos. En el año 1997, Resnick y Varian (11) proponen llamar a estos sistemas con el nombre de "sistemas de recomendación", dado que por esa fecha existían sistemas que no sólo se limitaban al filtro de información; sino que habían aparecido nuevos sistemas en el que no se utilizaban las opiniones de otros usuarios (11).

Con el trabajo "Diseminación selectiva de información" de Housman y Kaskela (12) apareció otra de las primeras formas de filtrado de información electrónica, en el que se diseñó un método que de forma automática se pudiera mantener a los científicos informados sobre nuevos documentos publicados en sus áreas de trabajo o especialización. El método se basaba en la creación de un perfil de usuario que contenía ciertas palabras clave relevantes para el usuario, que eran utilizadas para buscar coincidencias entre estas palabras clave y los nuevos documentos o artículos con el fin de intentar predecir qué artículos o documentos serían del interés de los científicos. Esta primera aproximación a los sistemas de recomendación fue implementada, pero su uso fue mucho menor de lo esperado (13).

A partir de entonces aparecieron nuevas ideas como la propuesta por Allen (14), en la que se consideraba la creación de modelos de usuario, con el objetivo de predecir qué nuevas publicaciones de artículos los científicos se prestarían a leer. El método, pese a mejorar los resultados de Packer y Soergel (13), era más efectivo prediciendo preferencias de usuario en categorías generales que de artículos específicos.

Otra aproximación a los sistemas de recomendación se dio con el sistema de información *Lens.* Este sistema estaba basado en el contexto del correo electrónico y permitía a los usuarios crear reglas para filtrar los correos electrónicos. Así pues, los usuarios podían, por ejemplo, crear una regla para etiquetar todos los mensajes provenientes de cierta persona o correo electrónico. El problema del sistema de información *Lens* apareció con las personas o usuarios cuyo

conocimiento en el ámbito de la informática era elemental o casi nulo. Estos usuarios eran incapaces de crear reglas de filtrado para priorizar o filtrar los correos recibidos (13).

Los primeros intentos de sistemas de recomendación desarrollados necesitaban una forma eficiente para mejorar la calidad de las recomendaciones. La calidad de la recomendación dependía en gran medida de la cantidad de información que fuera capaz de recopilar el sistema sobre los gustos y preferencias de los usuarios. Para mantener actualizado el sistema se emplean técnicas de retroalimentación.

1.2.2. Retroalimentación

Un sistema de recomendación debe evolucionar en el tiempo en cuanto a la calidad de sus recomendaciones. Para conseguir este objetivo se utilizan dos técnicas diferentes. Estas pueden ser adoptadas para almacenar la retroalimentación del usuario. La primera técnica llamada retroalimentación explícita es cuando el sistema requiere que el usuario explícitamente pondere los ítems. La otra técnica llamada retroalimentación implícita no requiere que el usuario activo se involucre porque la retroalimentación se obtiene de monitorear y analizar las actividades del usuario. (11)

La evaluación explícita indica cuán relevante o interesante puede ser un ítem para el usuario. Los tres enfoques para la retroalimentación explícita son los siguientes:

- Me gusta/No me gusta: los ítems son clasificados como relevantes o no relevantes adoptando una simple escala binaria de ponderación.
- Ponderaciones: una escala numérica discreta para evaluar los ítems. De forma alternativa se pueden utilizar símbolos que serán mapeados con la escala numérica donde los usuarios pueden evaluar los ítems mediante estrellas, círculos entre otros.
- Comentarios de textos: comentarios acerca de un ítem son recolectados y presentados a los usuarios para facilitar el proceso de toma de decisiones. Estos comentarios ayudan a decidir a los usuarios si un ítem ha sido apreciado o no por la comunidad. Presenta como desventaja que los usuarios deben leer e interpretar si los comentarios han sido positivos o negativos y en qué grado.

Los métodos de retroalimentación implícita se basan en asignar una puntuación de relevancia a las acciones de un usuario sobre un ítem tales como: guardarlo, imprimirlo, borrarlo, marcarlo para leer luego.

1.2.3. Resultados del estudio de antecedentes

El desarrollo de estas primeras herramientas sentó las bases para el futuro desarrollo de los sistemas de recomendación. Cada antecedente dio origen a un tipo de sistema de recomendación o a una estrategia de filtrado de información. Muchos de los sistemas utilizados actualmente emplean como base la experiencia adquirida de estos experimentos pioneros.

1.3. Clasificación de los sistemas de recomendación

Los diferentes enfoques para tratar el tema de los sistemas de recomendación dan origen a diferentes clasificaciones. Atendiendo a su funcionamiento estos se clasifican de la siguiente forma (15):

Sistemas de Recomendación Demográficos (SRD): se basan en la idea de que personas con unas características demográficas dadas (edad, sexo, nivel de educación, domicilio) tengan gustos similares a otras personas con características demográficas similares. Actualmente estos sistemas se aplican a dominios muy concretos, debido a la dificultad de disponer de los datos demográficos necesarios para su funcionamiento y a la preocupación de sus usuarios por su privacidad.

Sistemas de Recomendación Basados en Contenidos (SRBC): contienen información sobre las características de cada producto y se intentan extraer relaciones entre las características y las valoraciones de preferencia de un usuario. Para ello se construye un perfil de usuario a partir de los productos que este ha valorado y se utiliza para evaluar los productos no experimentados. Estos sistemas necesitan que el usuario valore una cantidad suficiente de productos para poder recoger sus gustos o necesidades, además no pueden realizar recomendaciones de productos inesperados, es decir productos distintos de los que el usuario ha valorado previamente.

Sistemas de Recomendación Colaborativos (SRC): realizan recomendaciones para un usuario concreto (usuario activo) utilizando información proporcionada por el resto de los usuarios que hay en el sistema. En estos sistemas no es necesario tener información sobre los productos que se van a recomendar, ya que de los mismos sólo se conocen las valoraciones de preferencia que los distintos usuarios han dado. Además la precisión del sistema mejora con el tiempo, mientras más usuarios hayan en el mismo y mientras más valoraciones compartan con el resto mejores serán sus resultados.

Sistemas de Recomendación Basados en Conocimiento (SRBConoc): utilizan una base de conocimientos que describe cómo los distintos productos satisfacen las necesidades de un usuario y en qué medida. De esta manera el sistema encuentra el producto o productos que se ajustan a las necesidades que el usuario ha especificado. Para realizar una recomendación con pocas cantidades de información sobre los usuarios, estos sistemas necesitan tener un conocimiento adicional sobre el entorno en el que se utilizan. Además de tener información sobre las características de los productos, necesitan información sobre las relaciones entre los mismos, su capacidad para satisfacer las necesidades de los usuarios, entre otras. Esto puede resultar una tarea costosa ya que se necesita realizar un proceso de ingeniería del conocimiento para obtenerla. La aplicación de un sistema de recomendación basado en conocimiento en distintos dominios (películas, carros, libros, música) no resulta trivial y requiere un gran esfuerzo para trasladarlo de un dominio a otro.

Sistemas de Recomendación basados en Utilidad (SRBU): la información que se tiene sobre el usuario es una función que el usuario define para otorgar un valor de utilidad a los productos del catálogo. Esta función de utilidad se define mediante la combinación de los valores de los atributos de los productos. El SRBU evalúa todos los productos del catálogo y recomienda los N productos con mayor valor de utilidad al usuario.

Sistemas de Recomendación Híbridos (SRH): constituyen una forma de mejorar los resultados de recomendación ya que emplean la hibridación de distintas técnicas. En una hibridación de dos sistemas de recomendación se pretende que el algoritmo resultante presente todas las ventajas por separado de dichos sistemas de recomendación y ninguna de las desventajas o problemas de los mismos.

1.4. Análisis para seleccionar el tipo de sistema de recomendación

El estudio de los SR ha sido ampliamente estudiado siendo los dos paradigmas principales los SRC y los SRBC.(16) Los SRBC tratan de recomendar ítems similares a aquellos ítems que le han gustado al usuario en el pasado y los SRC identifican usuarios cuyas preferencias son similares a aquellas que tiene el usuario activo y le recomienda los ítems que aquellos usuarios le han gustado (17). A continuación se definen las características de ambos SR y se describe cómo funciona el proceso de recomendación de cada uno de ellos.

1.4.1. Sistemas de Recomendación Basados en Contenido

Estos sistemas analizan un conjunto de ítems previamente evaluados por un usuario, y construyen un modelo o perfil de usuario basado en los intereses de los usuarios sobre las características de los ítems evaluados por ese usuario (18). El perfil de usuario es una representación estructurada de los intereses de los usuarios, adoptada para recomendar nuevos ítems interesantes. El proceso de recomendación básicamente consiste en hacer coincidir los atributos del perfil de usuario con los atributos del contenido de un ítem. El resultado es una valoración de cuan interesante le resultará al usuario ese documento. Los sistemas de filtrado de información basados en contenido necesitan técnicas propias para representar los ítems y producir el perfil de usuario, y algunas estrategias para comparar el perfil de usuario con la representación de los ítems. El proceso de recomendación es realizado en tres pasos, cada uno de ellos es manejado como un componente separado (16):

Analizador de contenido: es necesario cuando la información no tiene una estructura, realizar algún tipo de pre-procesamiento para extraer la información relevante de forma estructurada. La principal responsabilidad de este componente es representar el contenido del ítem que proviene de las fuentes de información en una forma que pueda ser entendible por las próximas fases del proceso. Una de las estrategias es transformar la información de contenido en vectores de palabras clave. Esta representación constituye la entrada del creador de perfil y del componente de filtrado.

Creador de Perfil: este módulo recolecta los datos representativos de las preferencias de los usuarios y trata de generalizar esta información, para construir el perfil de usuario. Esta generalización se realiza mediante el empleo de métodos de aprendizaje automatizado (19), las cuales son capaces de inferir un modelo con los intereses del usuario a partir de los ítems que este ha ponderado en el pasado.

Componente de filtrado: este módulo utiliza el perfil de usuario para sugerir documentos relevantes, realizando una comparación entre la representación del perfil y los ítems que pudieran ser recomendados. El resultado es una valoración obtenida mediante el empleo de medidas de similaridad (20), obteniendo una lista ponderada de ítems que pudieran ser interesantes.

1.4.2. Sistemas de Recomendación Colaborativos

Los SRC (también llamados como filtrado colaborativo) emplean las ponderaciones del usuario activo y las ponderaciones de los restantes usuarios del sistema. La idea principal es que la ponderación del usuario activo para un nuevo ítem, es similar a aquella que realiza otro usuario con similar gusto. Los métodos de filtrado colaborativo pueden ser divididos en dos grupos generales: los basados en vecindario (basado en memoria (21) o basado en heurística (22)) y los basados en modelo. Los métodos de filtrado colaborativo basado en vecindario almacenan las ponderaciones que los usuarios han realizado de los ítems y las emplean directamente para predecir las ponderaciones de los nuevos ítems. Existen dos estrategias en el primer grupo para realizar las predicciones conocidas como usuario-usuario e ítem-ítem. Los sistemas basados en usuario (23), (24), (25) evalúan el interés de un usuario por un ítem empleando las ponderaciones que para este ítem han realizado otros usuarios, llamados vecinos, teniendo estos vecinos patrones similares de evaluación. Los sistemas basados en ítem (5, 26) predicen la ponderación de un usuario para un ítem basándose en las ponderaciones de ese usuario para ítems similares. En este esquema, dos ítems son similares si varios usuarios del sistema han ponderado esos ítems de igual forma.

Por el contrario los basados en modelo emplean las ponderaciones para generar un modelo predictivo. La idea general es modelar las interacciones de los usuarios con los ítems. Este modelo es entrenado empleando la información disponible y posteriormente se emplea el modelo para predecir las ponderaciones de los usuarios para los nuevos ítems. El enfoque basado en modelos emplea, para recomendar ítems, algoritmos como Clustering Bayesiano (21), Análisis Semántico Latente (27) y Máquinas de Soporte Vectorial (3).

A continuación se establece una comparación entre los SRBC y los SRC con el objetivo de determinar cuál será utilizado como guía en la creación del Módulo Recomendaciones para el sistema REPXOS 3.0.

Tabla 1. Comparación entre los Sistemas de Recomendación basados en contenido y en el filtrado colaborativo(16)

	Sistemas de Recomendación Basados en Contenido (SRBC)	**Sistemas de Recomendación Colaborativos (SRC)**

Independencia del usuario: Capacidad que tiene el sistema para realizar recomendaciones basadas en las evaluaciones de los usuarios.	Explota únicamente las evaluaciones provenientes del usuario activo para construir su propio perfil.	Necesitan evaluaciones de otros usuarios a fin de encontrar los vecinos más cercanos del usuario activo, por lo que únicamente los ítems más gustados por los vecinos del usuario activo serán recomendados.
Transparencia: Claridad de los datos que son almacenados en el sistema por parte de los usuarios lo cual permite realizar recomendaciones confiables.	Mediante el listado explícito de las características que provocaron que un ítem apareciera en la lista de recomendaciones se puede realizar una explicación de cómo funciona el sistema de recomendación. Estas características son indicadores para consultar y poder determinar hasta qué punto confiar en una recomendación.	Solamente infieren la explicación de que un documento fue recomendado porque a un usuario desconocido con gustos similares le gusta ese documento.
Nuevo Ítem: Cuando un nuevo ítem es ingresado en el sistema.	Son capaces de recomendar ítems que no han sido evaluados por ningún usuario.	Padecen de la primera ponderación (necesitan que el nuevo documento sea evaluado por un número considerable de usuarios para poderlo recomendar)
Sobre-especialización: Capacidad que tiene el sistema de sorprender al usuario con las recomendaciones.	Estos sistemas solamente recomiendan ítems que se correspondan a los que están almacenados en el perfil del usuario, obteniendo como resultado recomendaciones muy similares, puesto a que estas siempre se basan en la misma información.	Tienen en cuenta las similitudes que pueden existir con otros usuarios, por lo que admiten sugerencias de elementos similares que los mismos han evaluado o considerado.

Nuevo usuario: Cuando un nuevo usuario se registra en el sistema.	Es necesario recoger suficientes ponderaciones de un usuario antes de que el sistema de recomendación pueda realmente entender las preferencias del usuario y brindar recomendaciones precisas. Si el usuario es nuevo, el sistema no será capaz de brindar recomendaciones confiables.	Un usuario debe de dar un número suficiente de valoraciones de preferencia para poder calcular su vecindario.

A partir de la **Tabla 1** se puede apreciar que el filtrado colaborativo resuelve algunas de las limitaciones del enfoque basado en contenido y viceversa. Aun así los sistemas de recomendación basados en el filtrado colaborativo son capaces de recomendar aquellos ítems cuyo contenido no está disponible o es de difícil obtención mediante la retroalimentación de otros usuarios. Además las recomendaciones que brindan se basan en la calidad de los ítems según fueron evaluados por otros usuarios. También los sistemas de filtrado colaborativo pueden recomendar ítems con contenido muy diferente por lo que las recomendaciones obtenidas como resultado no se basarán solo en la misma información contenida en el perfil del usuario. Esto proporciona a los usuarios, conocer ítems novedosos que nunca antes habían visto y que pueden contribuir al desarrollo de investigaciones científicas. Por sus ventajas se decide emplear el filtrado colaborativo para el desarrollo del Módulo Recomendaciones del sistema REPXOS 3.0.

La selección del filtrado colaborativo como guía para el desarrollo del Módulo Recomendaciones del sistema para repositorios digitales REPXOS 3.0 comprende dos tareas principales: la elección del enfoque y del esquema que serán utilizados.

Enfoque basado en memoria

Utiliza la base de datos completa para generar una predicción. El funcionamiento de este enfoque es el siguiente: se utilizan técnicas estadísticas para encontrar un conjunto de vecinos al usuario activo y posteriormente se utilizan una serie de algoritmos que combinan las preferencias de esta vecindad para realizar las predicciones y recomendaciones. Este enfoque es muy popular y

exitoso en la práctica pero sufre los problemas de escasez y escalabilidad por lo que se hizo necesaria la aparición del enfoque basado en modelo. (15)

Ventajas del enfoque basado en memoria (16):

- Simplicidad: los métodos basados en memoria son intuitivos y relativamente simples de implementar. En su forma más simple solo un parámetro (el número de vecinos usados en la predicción) requiere de perfeccionamiento.
- Justificabilidad: los métodos basados en memoria proveen una justificación concisa e intuitiva para las predicciones calculadas.
- Eficiencia: son métodos muy eficientes pues no requieren de costosas fases de entrenamiento. Además pese a que la fase de recomendación requiere un costo de tiempo superior a los basados en modelos, los vecinos más cercanos pueden ser precalculados de forma *offline*, brindando recomendaciones casi instantáneas. Almacenar esos vecinos requiere poca memoria, convirtiendo ese enfoque en escalable para aplicaciones comerciales con millones de usuarios y productos.
- Estabilidad: son poco afectados por la constante adición de usuarios, productos y ponderaciones. Una vez que la matriz de similaridad ha sido calculada, el sistema puede realizar recomendaciones sin necesidad de tener que ser reentrenado.

Desventajas del enfoque basado en memoria (15):

- Escasez: los sistemas de recomendación colaborativos necesitan una gran cantidad de datos, muchos usuarios puntuando muchos ítems similares para así poder calcular los grupos de vecinos y, en base a ellos, realizar las recomendaciones. Si la base de datos contiene pocos usuarios o pocas puntuaciones por parte de cada usuario, la matriz de puntuaciones será muy escasa y los cálculos de vecindad, predicción y recomendación no pueden ser realizados con la suficiente seguridad y exactitud obteniendo por lo tanto unas recomendaciones de baja calidad.
- Escalabilidad: los sistemas de recomendación colaborativos usan por norma general algoritmos de cálculo de los k vecinos más cercanos (kNN) para obtener la similaridad entre usuarios. Estos algoritmos son costosos computacionalmente y su coste crece linealmente cuanto mayor sea el número de usuarios y de ítems por lo que con base de

datos con millones de elementos, al aumentar el número de datos, el sistema sufrirá graves problemas de escalabilidad.

Enfoque basado en modelo:

Proporciona recomendaciones de ítems desarrollando primero un modelo (ya sea mediante redes bayesianas, clustering o modelos basados en reglas) de las puntuaciones de los usuarios sobre los ítems. No se utilizan técnicas estadísticas sino una aproximación probabilística que calcula el valor esperado de una predicción del usuario dados sus puntuaciones sobre otros ítems. (15)

Los modelos se desarrollan utilizando la minería de datos, algoritmos de aprendizaje automático para encontrar patrones basados en los datos de entrenamiento. Estos se utilizan para realizar predicciones de datos reales. Hay muchos algoritmos de filtrado colaborativo basados en modelos. Estos incluyen redes bayesianas, modelos de clustering, modelos de semántica latente tales como la descomposición de valor singular, el análisis semántico latente probabilístico, entre otros.

Ventajas del enfoque basado en modelo (15):

- Maneja los datos esparcidos mejor que los basados en memoria, esto ayuda con la escalabilidad al contar con grandes conjuntos de datos.
- Se mejora el rendimiento de la predicción.
- Da un fundamento intuitivo para las recomendaciones.

Desventajas del enfoque basado en modelo (15):

- La construcción del modelo es muy costosa en tiempo.
- Al crear el modelo se puede perder información útil debido a la reducción del mismo.
- Existen modelos que presentan dificultad para explicar las predicciones por la complejidad con la que fueron construidos.

Recientes investigaciones han tratado de demostrar la superioridad del enfoque basado en modelo a la hora de predecir las evaluaciones de los ítems (28, 29), sin embargo la precisión de la recomendación por sí sola no garantiza que esta sea efectiva y satisfactoria para el usuario. También se debe tener en cuenta la novedad de la predicción que brinda recomendaciones de ítems interesantes que de otra forma el usuario no hubiera encontrado. Precisamente el enfoque basado en memoria tiene en cuenta este elemento pues captura las asociaciones existentes en

los datos y pueden dar recomendaciones variadas que le brinden al usuario nuevas perspectivas de gustos similares mas no iguales.

Las características del enfoque basado en memoria resaltan sus ventajas respecto al basado en modelo por lo que el enfoque basado en memoria será utilizado para la creación del Módulo Recomendaciones del sistema para repositorios digitales REPXOS 3.0. El enfoque basado en memoria puede ser implementado utilizando dos esquemas, usuario-usuario e ítem-ítem, ambos serán descritos a continuación.

Para elegir entre la implementación de un esquema basado en usuario o de uno basado en ítem son considerados 3 criterios principales los cuales serán descritos a continuación (16):

- Precisión: la precisión de un sistema basado en memoria depende mayormente de la proporción entre el número de usuarios y de ítems presentes en el sistema. Por lo general en sistemas donde el número de usuarios es superior al número de ítems, es mejor emplear el esquema basado en ítem pues produce recomendaciones más precisas (30). Por el contrario sistemas con un número de usuarios menor que de ítems es más beneficioso emplear un esquema basado en usuario.
- Eficiencia: el tiempo de recomendación en línea es el mismo para los dos esquemas sin embargo cuando el número de usuarios excede el número de ítems, los sistemas basados en ítem requieren menor tiempo y memoria para calcular la matriz de similaridad, haciéndolos más escalables. Por el contrario en el caso de que el número de ítems sea mayor que el de usuarios ambos métodos tienen igual comportamiento.
- Estabilidad: la elección del esquema depende de la frecuencia y la cantidad de cambios de usuarios e ítems que se realizan en el sistema. Si la lista de ítems disponibles es casi estática en comparación con la de usuario en el sistema, es preferible emplear un esquema basado en ítem, pues la matriz de similaridad (contiene la similaridad entre los usuarios del sistema) puede ser calculada de forma infrecuente y aun así seguir siendo capaz de recomendar ítems a nuevos usuarios. Por el contrario si la lista de ítems está en constante cambio es mejor emplear un esquema basado en usuario pues ha demostrado ser más estable.

Teniendo en cuenta los criterios previamente vistos y que el sistema cuenta con un número de ítems superior al número de usuarios donde la lista de ítems está en constante cambio se decide

emplear el esquema basado en usuario. Se hace necesario realizar un análisis de sistemas existentes que empleen el filtrado colaborativo para comprobar su factibilidad como modelo para el desarrollo del Módulo Recomendaciones del sistema REPXOS 3.0.

1.5. Sistemas de Recomendación Colaborativos existentes

A continuación se describen características de algunos sistemas de recomendación que se basan en el filtrado colaborativo existentes a nivel nacional e internacional.

- ✓ **Amazon**

Utiliza su sistema de recomendaciones como herramienta de marketing en muchas campañas por correo electrónico y en la mayoría de las páginas de su sitio Web. Al hacer clic en "Sus Recomendaciones", los clientes acceden a un área donde pueden filtrar sus recomendaciones en función de una línea de productos, calificar productos recomendados, evaluar sus compras anteriores y ver cuáles son los artículos recomendados. Estas recomendaciones son individuales y personalizadas para cada cliente. Se trata, por tanto, de un sistema de filtrado colaborativo “ítem -ítem”, es decir, que son recomendaciones contextuales por productos. Para realizar la recomendación, comparan los productos que el usuario ha consultado o ponderado con los consultados o ponderados que son similares, para ello emplean el algoritmo vecinos más cercanos y como medida de similitud la basada en coseno. (31).

- ✓ **Google News**

Este sistema de recomendación trata acerca de la implementación de un servicio de recomendación colaborativo de noticias. Posee una alta tasa de rotación de su catálogo de artículos: los artículos son noticias que tienen un flujo constante. Las últimas noticias son las más interesantes. Google News solo utiliza retroalimentación implícita para construir un perfil de usuario, cuando éste hace clic sobre los temas. El sistema es puramente colaborativo y se basa en los ID de noticias e ID de los usuarios, de modo que un usuario está representado por la lista de noticias en el que hizo clic. El método basado en la similitud utiliza sólo el recuento de aquellas visitas que un mismo usuario efectúa sobre una misma noticia, al menos dos veces. (31)

- ✓ **VideoWeb**

Este es un sistema de recomendación colaborativo para la personalización del trato a los usuarios de la plataforma VideoWeb en la UCI. La plataforma VideoWeb es un sistema de trasmisión de

contenido multimedia que brinda la posibilidad de interactuar con todo tipo de archivos multimedia y artículos de contenidos mediante una interfaz amigable. Para lograr una mejor organización de los archivos multimedia que se publican se dividen en tipologías, estas pueden ser: videos, películas, documentales, etc. El algoritmo empleado para el desarrollo de este sistema fue el K-medias. (32)

- ✓ **Last.fm**

Es una red social, una radio vía Internet y además un sistema de recomendación de música que construye perfiles y estadísticas sobre gustos musicales, basándose en los datos enviados por los usuarios registrados. En la radio se puede seleccionar las canciones según las preferencias personales (de acuerdo a un algoritmo y a las estadísticas) o de otros usuarios. Un usuario de Last.fm puede construir un perfil musical usando dos métodos: escuchando su colección musical personal en una aplicación de música con un plugin de Audioscrobbler[2], o escuchando el servicio de radio a través de Internet de Last.fm, normalmente con el reproductor de Last.fm. Las canciones escuchadas son añadidas a un registro desde donde se calcularán los gráficos de barras de tus artistas y canciones favoritas, además de las recomendaciones musicales. Las recomendaciones son calculadas usando un algoritmo de filtrado colaborativo, así los usuarios pueden explorar una lista de artistas no listados en su propio perfil pero que sí que aparecen en otros usuarios con gustos similares. Last.fm también permite a los usuarios manualmente recomendar discos específicos a otros usuarios.

1.5.1. Valoración sobre los Sistemas de Recomendación de filtrado colaborativo existentes

El estudio realizado permitió conocer el funcionamiento del proceso de recomendación de algunos SRC existentes tanto a nivel nacional como internacional, lo cual sirvió de guía en la planificación del desarrollo del Módulo Recomendaciones. Además con el estudio realizado se determinó que no se puede hacer uso de ninguno de los SR antes mencionados, ya que las autoras del trabajo definieron las características propias que tendrá el Módulo Recomendaciones en dependencia de las necesidades que presentaba el sistema REPXOS 3.0. A continuación se

[2] Complemento opcional que contiene un conjunto de plugin que agregan funcionalidades al SR Last.fm para almacenar de forma implícita los gustos musicales de los usuarios y poder buscar las similitudes contenidas en otros perfiles.

exponen las causas que dieron paso a que no se utilizaran ninguno de los SR vistos anteriormente:

- El SR Amazon utiliza un enfoque basado en memoria y de este emplea el esquema ítem-ítem mientras que para el desarrollo de Módulo Recomendaciones del sistema REPXOS 3.0 el esquema a usar es usuario-usuario.
- El SR desarrollado para la plataforma de VideoWeb en la UCI utiliza el enfoque basado en modelo, mientras que el Módulo Recomendaciones propuesto para el sistema REPXOS 3.0 usa el enfoque basado en memoria.
- Google News solo utiliza retroalimentación implícita para construir un perfil de usuario. Además para buscar las similitudes utiliza sólo el recuento de aquellas visitas que un mismo usuario efectúa sobre una misma noticia en al menos dos veces. El Módulo Recomendaciones a desarrollar utiliza la retroalimentación explícita para obtener las preferencias del usuario, además recomienda ítems que han sido votados por usuarios similares al usuario activo.
- Last.fm construye perfiles y estadísticas sobre gustos musicales, basándose en los datos enviados por los usuarios registrados. Los datos enviados (explicados anteriormente) los obtiene de forma implícita mediante la interacción del usuario con el sistema. El Módulo Recomendaciones a desarrollar utiliza la retroalimentación explícita para obtener las preferencias del usuario.

Luego de realizar un estudio sobre los sistemas de recomendación de filtrado colaborativo existentes se procede a caracterizar la metodología, herramientas, tecnologías y lenguajes que se emplearán para el desarrollo del Módulo Recomendaciones.

1.6. Metodologías, tecnologías, herramientas y lenguajes utilizados

Proceso de Desarrollo Unificado: (RUP, por sus siglas en inglés). Es una metodología robusta que proporciona un comportamiento disciplinado para realizar la asignación de tareas y responsabilidades en una organización de desarrollo. RUP está dirigido por casos de uso, centrado en la arquitectura y es iterativo e incremental. El proceso Unificado se repite a lo largo de una serie de ciclos que constituyen la vida de un sistema. Cada ciclo consta de cuatro fases: inicio, elaboración, construcción y transición. Cada fase de subdivide en iteraciones. En cada iteración se desarrolla en secuencia un conjunto de disciplinas o flujos de trabajos las cuales son:

recopilación de requisitos, análisis, diseño, implementación y pruebas. Se utilizó con el objetivo de obtener un software eficiente, mediante la planificación total del trabajo a realizar. (33)

Lenguaje Unificado de Modelado: se emplea para la construcción de Modelos (UML, por sus siglas en inglés), constituye un estándar para el modelado de software, permite especificar, visualizar, construir y documentar los artefactos de un sistema de software. Es un sistema notacional destinado a los sistemas de modelado que utilizan conceptos orientados a objetos. Se empleó con el objetivo de visualizar el producto del trabajo (artefactos) en esquemas o diagramas estandarizados. (34)

Sistema Base: Dspace 4.2: es una herramienta para las bibliotecas digitales diseñada para capturar, almacenar, ordenar, conservar y redistribuir la producción intelectual y de investigación de una Institución en formato digital. Es un software de código abierto que provee herramientas para la administración de colecciones digitales y comúnmente es usada como solución de repositorio institucional. Soporta una gran variedad de datos, incluyendo libros, tesis, fotografías, videos, datos de investigación y otras formas de contenido. Los datos son organizados como ítems que pertenecen a una colección, cada colección pertenece a una comunidad. Dspace cuenta con algunas características de gran importancia que permiten y facilitan su uso en una gran cantidad de instituciones. Entre ellas se encuentra su distribución bajo Licencia Berkeley Software Distribution (BSD, por sus siglas en inglés); que es multiplataforma, por lo que se adapta a gran número de Sistemas Operativos; es basado en tecnología web y es un sistema adaptable (puede soportar el flujo y carga de trabajo); además posee un sistema de Handles (Manejadores) de archivos que brinda muchas posibilidades a los usuarios.(35)

Sistema Gestor de Base de Datos: PostgreSQL 9.3: es una base de datos relacional, distribuida bajo licencia Berkeley Software Distribution y con su código fuente disponible libremente. Entre sus características posee estabilidad, potencia, robustez, facilidad de administración e implementación de estándares. Además funciona con grandes cantidades de datos y una alta concurrencia, con muchos usuarios accediendo a la vez el sistema. Se utilizó con el objetivo de almacenar la información con la que trabaja el Módulo Recomendaciones.(36)

Herramientas de administración PgAdmin3: herramienta gráfica de código abierto para administrar y desarrollar bases de datos en PostgreSQL. Esta herramienta está diseñada para responder a las necesidades de la mayoría de los usuarios, desde escribir simples consultas SQL

hasta bases de datos complejas. Está disponible en varios lenguajes y para varios sistemas operativos. Se utilizó con el objetivo de trabajar con los objetos que están almacenados en la base de datos, examinar sus propiedades y realizar tareas administrativas. (37)

Entorno Integrado de Desarrollo: NetBeans 8.0: es un Entorno Integrado de Desarrollo o Integrated Development Environment (IDE, por sus siglas en inglés) en el cual se realizan todas las tareas asociadas a la programación: editar el código, compilarlo, ejecutarlo, depurarlo. Es modular, de base estándar (normalizado), escrito en el lenguaje de programación Java. El proyecto NetBeans constituye un IDE de código abierto y una plataforma de aplicación, las cuales pueden ser usadas como una estructura de soporte general (framework) para compilar cualquier tipo de aplicación. (38)

Contenedor Web: Apache Tomcat 7: es un contenedor web basado en el lenguaje Java que actúa como motor de servlets y JSPs desarrollado bajo el proyecto Jakarta en la Apache Software Foundation. Tomcat implementa las especificaciones de los servlets y de JavaServer Pages (JSP) de Oracle Corporation. Es desarrollado en un entorno abierto y participatorio, bajo la licencia de Apache Software License. Apache Tomcat es empleado en un gran número de aplicaciones como Dspace, CiteSeerX y JBoss. Se utilizó para el desarrollo y despliegue del sistema REPXOS 3.0.(39)

Herramienta de instalación Ant: es una herramienta usada en programación para la realización de tareas mecánicas y repetitivas, normalmente durante la fase de compilación y construcción (build). Es, por tanto, un software para procesos de automatización de compilación, similar a Make pero desarrollado en lenguaje Java y requiere la plataforma Java, así que es más apropiado para la construcción de proyectos Java. Esta herramienta se basa en archivos de configuración XML y clases Java para la realización de las distintas tareas, siendo idónea como solución multiplataforma. Ant es flexible y no impone convenciones de código a los proyectos Java, siendo adoptada como su herramienta de construcción. Se utilizó para la compilación y construcción del proyecto.(40)

Herramienta de compilación Maven 3: utiliza un Project Object Model (POM) para describir el proyecto de software a construir, sus dependencias de otros módulos y componentes externos, y el orden de construcción de los elementos. Viene con objetivos predefinidos para realizar ciertas tareas claramente definidas, como la compilación del código y su empaquetado. Es el estándar

actual para construir, administrar y automatizar aplicaciones en java. Es gratis y de código abierto, cuenta con una activa comunidad de soporte. Se utilizó para construir el proyecto y administrar sus módulos. (41)

Lenguaje de desarrollo Java: es un lenguaje de programación de uso general, orientado a objetos y multiplataforma. Gracias a la *Application Programmer Interface* (API)[3] de java se puede ampliar el lenguaje para que sea capaz de, por ejemplo, comunicarse con equipos mediante red, acceder a bases de datos, crear páginas HTML dinámicas, crear aplicaciones visuales al estilo Windows. Se utilizó en el proyecto como lenguaje de programación por sus ventajas y facilidades anteriormente descritas. (22, 23)

Conclusiones del capítulo

En el presente capítulo se evidencian los resultados del estudio realizado sobre los sistemas de recomendación y se arriban a las siguientes conclusiones:

- ✓ A partir del estudio de las tendencias de los SR existentes en la actualidad se determinó que los más usados son los SRC y los SRBC. Luego de comparar ambos sistemas, se decidió emplear un SRC por las ventajas que este ofrece para la creación del Módulo Recomendaciones.
- ✓ Después de analizar las características del filtrado colaborativo, se percibió que estos pueden ser divididos en dos enfoques: basados en memoria y en modelo. Al investigar las ventajas y desventajas de ambos enfoques, se decidió emplear el enfoque basado en memoria.
- ✓ Luego de realizar un estudio acerca del enfoque basado en memoria se percibió que este puede ser implementado mediante dos esquemas: usuario-usuario e ítem-ítem. Teniendo en cuenta las características de cada esquema, se decidió emplear el esquema usuario-usuario por ser el que más se ajusta a las características presentes en el Módulo Recomendaciones a implementar.

[3] La Interfaz de Programación de Aplicaciones (IPA), abreviada como API (del inglés: Application Programmer Interface) es el conjunto de funciones y procedimientos ofrecidas por una biblioteca informática para ser utilizados por otro software como una capa de abstracción.

- ✓ Después de realizar una valoración acerca de algunos SRC se decidió no emplear ninguno ya que los anteriormente vistos no se ajustan a las necesidades del sistema REPXOS 3.0.

CAPÍTULO 2. **CARACTERÍSTICAS DEL MÓDULO RECOMENDACIONES DEL SISTEMA PARA REPOSITORIOS DIGITALES REPXOS 3.0**

El presente capítulo tiene como objetivo describir la propuesta del Módulo Recomendaciones para el sistema REPXOS 3.0, enfocado a contribuir a la recuperación de la información más relevante para los usuarios según su criterio. A partir del flujo descrito se realiza el Modelo de Dominio correspondiente, para luego extraer las características con las que contará el Módulo Recomendaciones a desarrollar. Se listan los requerimientos funcionales y no funcionales, se definen los actores y los casos de uso del sistema, mostrándose un breve resumen de cada uno.

2.1. Modelo de Dominio

Un modelo del dominio es una representación de las clases conceptuales del mundo real, no de componentes software. No se trata de un conjunto de diagramas que describen clases software, u objetos software con responsabilidades. (34)

Al no ser identificado claramente los procesos del negocio, solo elementos conceptuales, se decide realizar la representación de estos elementos a través un modelo de dominio. Con ello se pretende lograr una mejor compresión de cómo ocurren los procesos, dando paso a la identificación de los requisitos funcionales. A continuación se muestra el modelo donde se describen los principales conceptos relacionados con el sistema, así como las relaciones entre ellos:

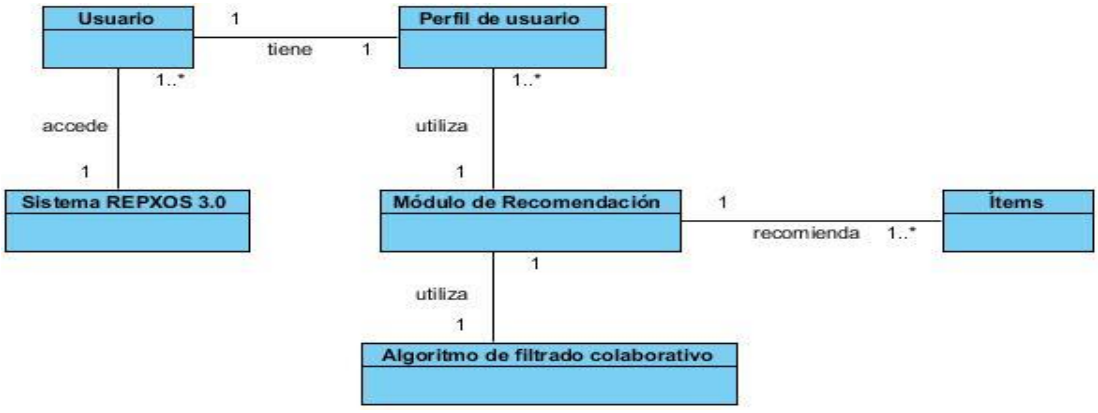

Figura 1. Modelo de Dominio del Módulo Recomendaciones.

Definición de los conceptos del modelo del dominio:

Algoritmo de filtrado de colaborativo: algoritmo que utiliza el módulo para realizar el filtrado de la información.

Ítems: constituyen los ítems que el sistema o usuarios recomiendan.

Módulo de Recomendación: es el encargado de mostrarle al usuario la lista de ítems recomendadas por el propio sistema o por otros usuarios.

Perfil de usuario: es donde se almacenan los datos y preferencias de usuarios.

Sistema REPXOS 3.0: sistema que facilita y automatiza los procesos y servicios que brinda el repositorio a sus usuarios.

Usuario: es la persona que interactúa con el sistema y sus funcionalidades.

2.2. Especificación de los requisitos de software

La captura de los requisitos es una de las actividades fundamentales que se desarrollan durante el proceso de construcción de un software. Esta constituye las condiciones o capacidades que el software debe cumplir, por tanto, es necesario que sean verificados y validados para evitar errores en futuras etapas que conllevarán a resultados inesperados y atrasos en el desarrollo.

2.2.1. Requisitos Funcionales

Son enunciados acerca de servicios que el sistema debe proveer, de cómo debería reaccionar el sistema a entradas particulares y de cómo debería comportarse el sistema en situaciones específicas. (44)

- ✓ **RF1:** Ponderar material
- ✓ **RF2:** Recomendar material
- ✓ **RF3:** Buscar usuario a recomendar
- ✓ **RF4**: Visualizar recomendación de material a usuario
- ✓ **RF5**: Eliminar recomendación de material a usuario
- ✓ **RF6**: Añadir a la lista los documentos consultados

La siguiente tabla muestra la descripción de los requerimientos funcionales identificados para el desarrollo del Módulo Recomendaciones:

Tabla 2. Requerimientos funcionales del Sistema

Requerimiento	Descripción
RF 1. Ponderar material.	Se evalúa un material, según la apreciación del usuario, en una escala del 1 al 5, donde 5 es la máxima calificación. Además se calcula el promedio de las ponderaciones de ese material.
RF 2. Recomendar material.	Permite al usuario recomendar un documento a otros usuarios.
RF 3. Buscar usuario a recomendar.	Permite buscar al usuario que se le desea recomendar un documento.
RF 4. Visualizar recomendación de material a usuario.	Muestra las recomendaciones realizadas al usuario ya sea de forma automática por el sistema o manual.
RF 5. Eliminar recomendación de material a usuario.	Permite al usuario eliminar aquellas recomendaciones que no considere importante.
RF 6. Añadir a la lista los documentos consultados.	Añade a la lista de documentos consultados un nuevo documento.

2.2.2. Requisitos no funcionales

Es importante especificar las funcionalidades que debe desempeñar un sistema, así como definir todas las cualidades que lo hacen más atractivo al cliente, usable, rápido y confiable. Los requisitos no funcionales se reconocen como un conjunto de características de calidad, que es necesario tener en cuenta al diseñar e implementar el software.

La tabla muestra los requisitos no funcionales del sistema:

Tabla 3. Requerimientos no Funcionales del Módulo

Prefijo	Requerimiento	Descripción
Usabilidad	RNU 1. Facilidad de empleo para usuarios sin experiencia	El sistema debe tener una interfaz de fácil aprendizaje para que usuarios inexpertos puedan familiarizarse rápidamente. Se documenta el sistema con un manual de usuario con el objetivo de explicar su uso.
Usabilidad	RNU 2. Utilizar patrón de navegación	El sistema permitirá el fácil acceso a las funcionalidades del mismo. Además, se deben brindar accesos directos a las funcionalidades más utilizadas, mediante menú local y global en el sistema.
Eficiencia	RNE 1. Capacidad	El sistema debe soportar al menos 100 usuarios conectados.
Diseño e Implementación	RNDI 1. Lenguajes de programación	Se utiliza para la construcción del sistema los lenguajes de programación Java, HTML, JavaScript y las herramientas que se utilizan son de distribución bajo licencias libres, para favorecer el desarrollo de nuevos módulos o subsistemas.
Diseño e Implementación	RNDI 2. Forma de acceso	Al sistema se puede acceder a través de un navegador web desde los sistemas operativos Windows y GNU/Linux.

Diseño e Implementación	RNDI 3. Entorno Integrado de Desarrollo	Se utilizará NetBeans como entorno integrado de desarrollo. Dicho IDE soporta Java como lenguaje de programación.
Interfaz	RNI 1. Interfaces de usuario	Para acceder al Sistema debe usarse una versión del navegador Mozilla/Firefox v 35.0. No se garantiza la correcta visualización en otros navegadores. Además, que para acceder al repositorio el navegador debe tener habilitado el soporte para Java Script.
Seguridad	RNS 1. Protección de los datos almacenados	El sistema debe garantizar la protección de los datos almacenados. Para ello se establecerán diferentes niveles de acceso, garantizando que cada usuario acceda solamente a las funcionalidades permitidas. En el Módulo Recomendaciones esto se logra mediante la correcta configuración de web.xml, encargado de controlar el acceso.
Funcionamiento	RNFO 1. Requisitos de Software	El sistema debe integrarse con el Gestor de base de datos PostgreSQL 9.3, contenedor web utilizado como servidor web Apache Tomcat 7.0 y herramienta de compilación Maven 3.0.
Funcionamiento	RNFO 2. Requisitos de Hardware	El hardware donde se instalará el sistema debe poseer al menos una Interfaz de red cuya velocidad de transferencia iguale o supere los 100 Mbps. El Servidor donde se instale el Sistema debe tener como mínimo un Microprocesador Intel-Pentium Dual Core 2.0

		Mhz3 o equivalente, 4 GB de memoria RAM y un espacio libre en Disco Duro de 500 GB. (Se necesita espacio adicional para el almacenamiento de los documentos digitales).

2.3. Propuesta de solución

El proceso de recomendación de documentos digitales contenidos en el sistema REPXOS 3.0, permite recomendar a los usuarios del sistema materiales de acuerdo a su interés. El módulo propuesto permitirá a los usuarios del sistema REPXOS 3.0 la posibilidad de que, una vez autenticados, puedan evaluar los ítems consultados. La retroalimentación utilizada será la explícita. Para realizar las evaluaciones de los ítems del sistema se empleará una escala numérica discreta del 1-5, en dependencia de la preferencia que tenga el usuario sobre dicho material. Como alternativa, la escala numérica se representará mediante estrellas, donde cada una de ellas tendrá asociado un valor dentro de la escala definida. El objetivo es que esta información sea usada por el propio sistema para recomendar de forma automática los ítems con más alto nivel de ponderación. El módulo permitirá además, que se realicen recomendaciones entre usuarios, brindando así, un área de intercambio de información. Los usuarios tendrán la posibilidad de acceder a la lista de documentos que le han sido recomendados, ya sea de forma automática o las realizadas por otros usuarios.

Como estrategia de desarrollo se tendrá en cuenta que el Módulo Recomendaciones a implementar se basa en un SR de filtrado colaborativo empleando un enfoque basado en memoria. De este enfoque se utilizará el esquema usuario-usuario que tiene como objetivo recomendarle al usuario activo, aquellos ítems que han ponderado sus vecinos más similares y que él no ha ponderado. A continuación se analizaron las etapas por las que transita un SRC basado en memoria, estas son: la normalización de los ratings, medida de similaridad y selección de vecinos.

2.3.1. Etapas de un sistema de recomendación basado en memoria

Un sistema de recomendación basado en memoria comprende 3 etapas importantes. La primera etapa es la **normalización de los ratings (votos)**, la segunda etapa es el **cálculo de la matriz de similaridad de peso** y la última etapa es la **selección de los vecinos**. Para poder explicar de manera formal la tarea de recomendación, es necesario introducir algunas notaciones. Teniendo en cuenta de que un usuario $u \in U$ no

puede ponderar un ítem $i \in I$ más de una vez de forma r_{ui}. (11). A continuación se presenta una tabla que contiene las notaciones de las variables.

Tabla 4. Notación de las variables presentes en las fórmulas

u	**Usuario**
i	Ítem
U	Conjunto de usuarios
I	Conjunto de ítems
R	Conjunto de ponderaciones almacenadas en el sistema
S	Conjunto de posibles valores que puede tomar una ponderación $S = ([1 \dots,5])$
U_i	Subconjunto de usuarios que han ponderado el ítem i
I_u	Subconjunto de ítems que han sido ponderados por un usuario u
I_{uv}	Ítems que han sido ponderados por dos usuarios u y v
U_{ij}	Conjunto de usuarios que han ponderado los ítems i y j .
W_{uv}	Similaridad de preferencia entre usuario u y v
r_{ui}	Ponderación realizada por un usuario u para un ítem i
r_{vi}	Ponderación realizada por un usuario v para un ítem i
$\bar{r}_u$	Media de todos los votos para el usuario u
$Ni(u)$	K usuarios más semejantes a u que han ponderado i
v_{ir}	Voto dado por los KNN de u para el rating r

Primera etapa

La normalización de los rating se emplea para resolver el problema que existe por las diferentes apreciaciones que los usuarios tienen a la hora de ponderar los ítems. Dos de las más populares técnicas de normalización de ratings que han sido propuestas son Centrado por medias o Z puntuación normalización.

Centrado por medias

La idea de centrado por medias es determinar si una ponderación es positiva o negativa comparándola con la ponderación promedio. En el esquema basado en usuario, una ponderación r_{ui} se transforma mediante centrado por medias $h(r_{ui})$, restándole a r_{ui} el promedio r_u de las ponderaciones dadas por el usuario u a los ítems I_u.

$$h(r_{ui}) = r_{ui} - \bar{r}_u$$

***Fórmula 1**. Cálculo de la técnica de normalización Centrado por Medias* (16)

Una propiedad interesante de centrado por medias es que puede verse en el signo si la apreciación de un usuario por un ítem es positiva o negativa.

Z puntuación normalización

Z puntuación aparte de remover los valores anómalos causados por las diferentes percepciones, considera el esparcimiento en escalas individuales de ponderación. En el esquema basado en usuario la normalización divide la centrada por medias entre la desviación estándar de las ponderaciones dadas por el usuario.

$$h(r_{ui}) = \frac{r_{ui} - \bar{r}_u}{\sigma_u}$$

***Fórmula 2**. Cálculo de la técnica de normalización Z puntuación* (16)

Configuración de la primera etapa

Z puntuación normalización es mayormente utilizada para escalas continuas y Centrado por Medias para escalas discretas. En Z puntuación normalización como los valores son multiplicados por valores diferentes de desviaciones estándar, puede ser más sensible a valores anómalos.

En esta etapa, las autoras del presente trabajo deciden seleccionar de las técnicas antes mencionadas, para normalizar, la técnica Centrado por Medias ya que el Módulo Recomendaciones a desarrollar trabajará con una escala de valores discretos. Esta etapa se corresponde con el preprocesamiento de los datos desarrollado por el algoritmo k Vecinos más Cercanos.

Segunda etapa

Este es uno de los aspectos más críticos para construir un sistema de recomendación basado en memoria, esta etapa tiene un impacto significativo en la precisión y el rendimiento. Entre las medidas de similaridad más conocidas se encuentran la Correlación de Pearson y la del Coseno. A continuación se muestran las dos métricas de similaridad propuestas que básicamente lo que hacen es calcular las distancias que hay entre cada par de usuarios.

Coeficiente de correlación de Pearson

Es una métrica típica de similitud entre funciones de preferencias de usuarios o (menos frecuentemente) distancias de vectores o productos puntos. La fórmula da una aproximación de qué tan bien los valores comparados (perfiles, ítems) coinciden en la escala desde 0 (no similares) a 1 (total coincidencia) ó -1 (total diferencia). Este método indica cuan linealmente correlacionados están los elementos de un vector con los de otro.

$$PC(u,v) = \frac{\sum_{i \in I_{uv}} (r_{ui} - \bar{r}_u) - (r_{vi} - \bar{r}_v)}{\sqrt{\sum_{i \in I_{uv}} (r_{ui} - \bar{r}_u)^2 \sum_{i \in I_{uv}} (r_{vi} - \bar{r}_v)^2}}$$

***Fórmula 3**. Cálculo de la similitud basada en Correlación de Pearson* (16)

Similitud basada en coseno

Se considera cada elemento como un vector dentro de un espacio vectorial de m dimensiones y se calcula la similitud como el coseno del ángulo que forman. Es decir si se tienen dos vectores x_u, x_v consistentes en un arreglo cuyos elementos son las votaciones recibidas de cada usuario.

$$CV(u,v) = \cos(x_u, x_v) = \frac{\sum_{i \in I_{uv}} r_{ui} r_{vi}}{\sqrt{\sum_{i \in I_u} r_{ui}^2 \ \sum_{j \in I_v} r_{vj}^2}}$$

***Fórmula 4**. Cálculo de la similitud basada en coseno* (16)

Configuración de la segunda etapa

La medida de similitud a emplear para calcular la similitud entre usuarios es el Coeficiente de Correlación de Pearson. Esta medida da una aproximación de qué tan similares puede ser un usuario del sistema con respecto a otro, en el momento de realizar las recomendaciones. Este coeficiente captura las correlaciones negativas entre las preferencias de los usuarios y la apreciación de los diferentes ítems. El contar con estas correlaciones negativas puede incrementar la precisión de las predicciones (45). Además ha sido demostrado que el Coeficiente de Correlación de Pearson ha sido reconocida como la mejor medida de similitud para emplear en los sistemas de recomendación basado en memoria (45). Esta etapa se corresponde con la selección de la medida de similitud del algoritmo k Vecinos más cercanos.

Tercera Etapa

El número de vecinos cercanos para seleccionar y el criterio usado para la selección afectan la calidad del sistema de recomendación. Esta se divide en dos pasos: prefiltrado de los vecinos y por predicción.

Prefiltrado de los vecinos

Para no almacenar en la memoria toda la matriz de similaridad se plantean varias estrategias.

- Mejores N- Para cada usuario se calcula una lista de los n vecinos más cercanos y su respectiva similaridad es guardada.
- Filtrado por umbral- En vez de almacenar un número fijo de vecinos se mantienen los vecinos cuya similaridad es mayor a un determinado umbral. Es mejor que la anterior pues solo los vecinos más significativos son almacenados.
- Filtrado negativo- Eliminar las similitudes negativas.

Predicción.

Una vez que la lista de vecinos candidatos ha sido calculada para cada usuario, la predicción de la nueva ponderación se hace normalmente con el algoritmo k Vecinos más Cercanos, lo que significa que los k vecinos cuya similaridad es la mayor serán elegidos. El valor de k es uno de los parámetros más difíciles de configurar y es necesario aplicar prueba y error. Una variante que se utiliza cuando se premia la novedad por encima de la precisión es encontrar el usuario más similar al activo y recomendarle el ítem que este haya ponderado como el mayor.

Para realizar la predicción el esquema usuario-usuario del enfoque basado en memoria, lo hace mediante Regresión o Clasificación.

Regresión

Predicen el rating r_{ui} usando los ratings que han dado usuarios similares a u al ítem i, llamados vecinos más cercanos. Por cada usuario $v \neq u$ se tiene un valor W_{uv}. Los k Vecinos más cercanos de u son denotados por N_u, son los k usuarios v con la similaridad W_{uv} más alta. Sin embargo solo los usuarios que a su vez hayan ponderado el ítem i pueden ser utilizados en la predicción de r_{ui} y por tanto solo serán considerados los k usuarios más semejantes a u que han ponderado i $Ni(u)$. El rating $\hat{r}_{ui}$ puede ser estimado como el rating promedio dado a i por los vecinos de u.

$$\hat{r}_{ui} = \frac{1}{|N_i(u)|} \sum_{v \in N_i(u)} r_{vi}$$

***Fórmula 5**. Cálculo de la predicción usando Regresión* (16)

Esta fórmula no tiene en cuenta el hecho de que los vecinos pueden tener distintos niveles de similaridad. Una solución es pesar las contribuciones de cada vecino por su similaridad a u. Si estos pesos no suman 1, la ponderación estimada puede estar fuera del rango de valores permitidos. Por lo que se hace necesario normalizar estos pesos, y la ponderación promedio queda de la forma

$$\hat{r}_{ui} = \frac{\sum_{v \in N_i(u)} w_{uv} \; r_{vi}}{\sum_{v \in N_i(u)} |w_{uv}|}$$

***Fórmula 6**. Cálculo de la predicción usando Regresión teniendo en cuenta los distintos niveles de similaridad* (16)

Esta ecuación no tiene en cuenta el hecho de que los usuarios pueden usar diferentes valores de ponderación para cuantificar el mismo nivel de apreciación para un ítem. Este problema se resuelve normalizando los valores de ponderación de los vecinos. El resultado es necesario llevarlo a la escala original invirtiendo la normalización;

$$\hat{r}_{ui} = h^{-1}\left(\frac{\sum_{v \in N_i(u)} w_{uv} \; h(r_{vi})}{\sum_{v \in N_i(u)} |w_{uv}|}\right)$$

***Fórmula 7**. Cálculo de la predicción usando Regresión y normalizando* (16)

Clasificación

Por el contrario la clasificación en el esquema basado en usuario encuentra la ponderación más probable de un usuario u a un ítem i, calculando el voto de los vecinos de u. El voto v_{ir} dado por los kNN de u para el rating r puede ser obtenido como la suma de la similaridad de los pesos de los vecinos que han ponderado el ítem i.

$$v_{ir} = \sum_{v \in N_i(u)} \delta(r_{vi} = r) w_{uv}$$

***Fórmula 8**. Cálculo de la predicción usando Clasificación* (16)

Donde $\delta(r_{vi} = r)$ es 1 si $R_{vi} = r$ y 0 en caso contrario. Una vez que esto ha sido calculado para cada posible valor de ponderación, la ponderación estimada es simplemente el valor r para el cual v_{ir} fue el mayor valor.

Este método de clasificaciones puede ser utilizado con ponderaciones normalizadas y puede ser definido como:

$$\hat{\boldsymbol{r}}_{\boldsymbol{ui}} = \boldsymbol{h}^{-1}\left(\boldsymbol{arg\ max}_{\boldsymbol{r} \in \boldsymbol{S'}} \sum_{\boldsymbol{v} \in \boldsymbol{N_i(u)}} \boldsymbol{\delta}(\boldsymbol{h}(\boldsymbol{r_{vi}}) = \boldsymbol{r})\, \boldsymbol{w_{uv}}\right)$$

***Fórmula 9** Cálculo de la predicción usando Clasificación y normalizando* (16)

Configuración de la tercera etapa

De las estrategias planteadas en el prefiltrado de vecinos se utilizará:

- Mejores N- Para cada usuario del sistema se calcula una lista de los 5 vecinos más cercanos. La elección del valor 5 depende fundamentalmente de los datos. Para evitar problemas de eficiencia o precisión el valor fue seleccionado muy cuidadosamente. Se tuvo en cuenta que si N es muy largo se requería una gran cantidad de memoria para almacenar las listas de vecinos. Por el contrario si N era seleccionado muy pequeño podía reducir el campo del sistema de recomendación, provocando que algunos ítems nunca fueran recomendados.

Para realizar la predicción se utilizará la Clasificación, donde las autoras del presente trabajo decidieron realizarla de la siguiente forma:

Luego de seleccionar los 5 Vecinos más Similares al usuario activo, se procede a predecir los posibles votos que realizaría el usuario activo sobre los ítems que no ha votado. La predicción se realizará a través de la clasificación empleando el voto por mayoría. El objetivo de la clasificación es obtener un conjunto de predicciones para los ítems que el usuario activo no ha ponderado a partir de las votaciones de sus Vecinos más Similares. En base a estos valores se ordenan los ítems que se tendrán en cuenta para recomendar. Luego para mejorar el resultado de las recomendaciones, se realiza un segundo nivel del filtrado o clasificación basado en las coincidencias de las palabras clave que presentan estos ítems con las definidas en el perfil del usuario, aplicando además el mismo criterio, con respecto a los autores.

En caso de que no existan vecinos similares al usuario activo se realiza la recomendación buscando las coincidencias entre las palabras clave almacenadas en el perfil del usuario y aquellas contenidas en los metadatos de los ítems. Permitiendo que se le realicen recomendaciones a usuarios nuevos en el sistema que no tienen vecinos cercanos.

En la tercera etapa el prefiltrado de los vecinos se utiliza para optimizar el cálculo de las predicciones. La predicción comprende la selección de los k vecinos más similares y la aplicación de la función de evaluación del algoritmo k Vecinos más Cercanos.

2.4. Definición de los actores

Un actor es toda entidad externa al sistema que guarda una relación con el mismo y que le demanda una funcionalidad. Esto incluye a los operadores humanos, los sistemas externos y entidades abstractas como el tiempo. Un actor representa un rol en el sistema, no un usuario en específico del mismo.(34)

En el Módulo Recomendaciones solo interactúa un actor, el cual se define a continuación en la Tabla:

Tabla 5. Actores del Módulo Recomendaciones.

Actor	Descripción

Usuario	✓ Persona que interactúa con el sistema. ✓ Utiliza todas las funcionalidades que brinda el Módulo Recomendaciones. ✓ Consulta la información que le ha sido recomendada, ya sea de forma automática o hecha por otros usuarios. ✓ Recomienda ítems a otros usuarios.

2.5. Definición de los Casos de Uso del Módulo Recomendaciones

A partir del análisis de los requisitos funcionales del sistema se definieron los siguientes Casos de Uso:

- ✓ **CU1:** Ponderar material
- ✓ **CU2:** Recomendar material
- ✓ **CU3:** Buscar usuario a recomendar
- ✓ **CU4.** Gestionar recomendación de material a usuarios
- ✓ **CU5.** Añadir a la lista los documentos consultados

2.5.1. Diagrama de Caso de Uso del sistema

Los diagramas de casos de uso documentan el comportamiento de un sistema desde el punto de vista del usuario. Por lo tanto los casos de uso determinan los requisitos funcionales del sistema, es decir, representan las funciones que un sistema puede ejecutar. Su ventaja principal es la facilidad para interpretarlos, lo que hace que sean especialmente útiles en la comunicación con el cliente. (34)

El Diagrama de casos de uso para el Módulo Recomendaciones se muestra en la siguiente figura:

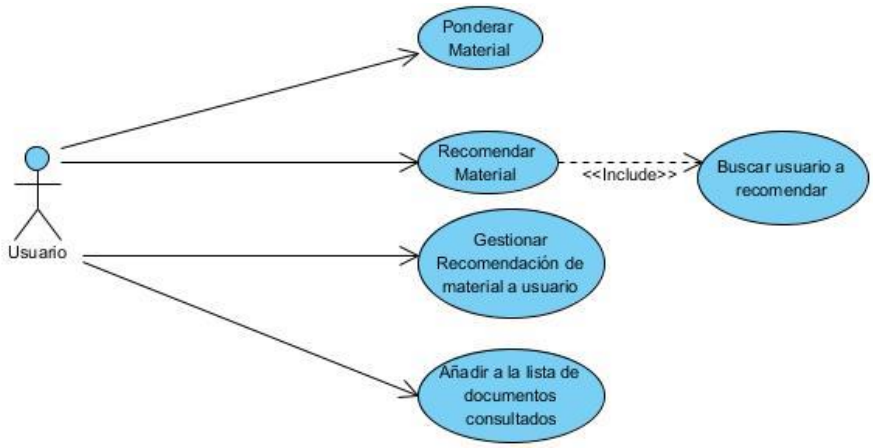

Figura 2. Diagrama de Casos de Uso del Módulo Recomendaciones

2.5.2. Descripción de los Casos de Uso del sistema

Para una mejor comprensión de los casos de uso se realiza la descripción de los mismos. A continuación se muestran las tablas que contienen las descripciones:

Tabla 6. Caso de Uso Ponderar Material

Objetivo	Evaluar un material consultado.
Actores	Usuario
Resumen	El caso de uso se inicia cuando el usuario decide evaluar un material consultado. El sistema permite dar un valor en una escala del 1 al 5. Termina así el caso de uso.
Complejidad	Baja
Prioridad	Baja
Precondiciones	El sistema debe estar instalado y ejecutándose correctamente. El usuario se ha autenticado en el sistema. El usuario tiene seleccionado un ítem Al menos en el sistema debe estar registrado un material.
Postcondiciones	Se evaluó el material.

Referencias	RF1

Tabla 7. Caso de Uso Recomendar Material

Objetivo	Recomendar material de usuario a usuario.
Actores	Usuario.
Resumen	Permite realizar la recomendación de material a un usuario de forma manual
Complejidad	Baja.
Prioridad	Baja.
Precondiciones	El sistema debe estar instalado y ejecutándose correctamente. El usuario se ha autenticado en el sistema. Al menos en el sistema debe estar registrado un material.
Postcondiciones	Se recomendó un material.
Referencias	RF2

Tabla 8. Caso de Uso Buscar Usuario a Recomendar

Objetivo	Buscar el usuario al que se le desea realizar la recomendación
Actores	Usuario.
Resumen	El caso de uso se inicia cuando el usuario que está actualmente autenticado en el sistema desea realizar la búsqueda del usuario al que le desea recomendar un material.
Complejidad	Baja.
Prioridad	Baja.
Precondiciones	El sistema debe estar instalado y ejecutándose correctamente. El usuario se ha autenticado en el sistema. Al menos en el sistema debe estar registrado un material.
Postcondiciones	Se buscó el usuario a recomendar

Referencias	RF3

Tabla 9. Caso de Uso Gestionar Recomendación de Material a Usuarios

Objetivo	Consultar y eliminar recomendación de material a un usuario
Actores	Usuario
Resumen	El caso de uso se inicia cuando el usuario desea consultar la lista de materiales recomendados por otros usuarios o por el propio sistema. El sistema permite visualizar el contenido de los materiales presentes en la lista y si lo desea puede eliminar el material que considere que no es de relevancia para su investigación. Termina así el caso de uso.
Complejidad	Baja
Prioridad	Baja
Precondiciones	El sistema debe estar instalado y ejecutándose correctamente. El usuario se ha autenticado en el sistema. Al menos en el sistema debe estar registrado un material
Postcondiciones	Se consultó o eliminó una recomendación
Referencias	RF4, RF5

Tabla 10. Caso de Uso Gestionar Lista de Documentos Consultados

Objetivo	Añadir documentos consultados por el usuario
Actores	Usuario
Resumen	El caso de uso se inicia cuando el usuario desea añadir un material a la lista de documentos consultados. Si el usuario selecciona la opción de añadir, el sistema añade este documento a la lista de consultados. Termina así el caso de uso.
Complejidad	Baja

Prioridad	Baja
Precondiciones	El sistema debe estar instalado y ejecutándose correctamente. El usuario se ha autenticado en el sistema. Al menos en el sistema debe estar registrado un material. El usuario tiene seleccionado un ítem
Postcondiciones	Se añadió a la lista de los documentos consultados.
Referencias	RF6

Conclusiones del capítulo

Con la realización de este capítulo se arribaron a las siguientes conclusiones:

- El SRC basado en memoria o vecindario que se utilizará emplea el algoritmo k Vecinos más Cercanos.
- A partir del estudio de las etapas por las que transita un SRC basado en memoria se definió para realizar la normalización emplear la técnica **Centrado por Medias** para la 1era etapa.
- Para la 2da etapa se definió emplear como medida de similitud **Correlación de Pearson**.
- Para la 3era etapa encargada de la selección de los vecinos, primeramente se aplicará la técnica de filtrado **Mejores N** y para realizar la predicción se aplicará como estrategia la **Clasificación**.

CAPÍTULO 3. **ARQUITECTURA Y DISEÑO DEL MÓDULO RECOMENDACIONES DEL SISTEMA PARA REPOSITORIOS DIGITALES REPXOS 3.0**

En este capítulo se propone el diseño del Módulo Recomendaciones para el sistema REPXOS 3.0, a partir de los Diagramas de Clases del Diseño empleando estereotipos web, los diagramas de colaboraciones, el diagrama de paquetes y el diseño de la base de datos. Se describen los elementos de estos diagramas, así como el estilo arquitectónico y los patrones de diseño utilizados en el Módulo Recomendaciones.

3.1. Modelo de diseño

El modelo de diseño constituye el conjunto de diagramas que describen el diseño lógico de un sistema. Se centra en los impactos que producen en el sistema a desarrollar los requerimientos funcionales y no funcionales. Comprende los diagramas de clases del diseño, diagramas de interacción, diagramas de paquetes, ofreciendo una perspectiva de especificación o implementación, como quiere el modelador. Es un modelo de objetos que describe la realización física de los casos de uso centrándose en como los requisitos funcionales y no funcionales, junto con otras restricciones relacionadas con el entorno de implementación, tienen impacto en el sistema a considerar. Además sirve de abstracción de la implementación del sistema y es, de ese modo, utilizado como una entrada fundamental de las actividades de implementación. (33)

3.1.1. Diagrama de Paquetes

Es una simplificación de un diagrama de clases, sólo se representan los paquetes a los que pertenecen las clases. Son muy útiles para ver las dependencias entre los paquetes de un sistema. Muestra cómo un sistema está dividido en agrupaciones lógicas mostrando las dependencias entre esas agrupaciones. Además suministran una descomposición de la jerarquía lógica de un sistema. Los Paquetes están normalmente organizados para maximizar la coherencia interna dentro de cada paquete y minimizar el acoplamiento externo entre los paquetes. (33)

Está constituido por dos tipos de elementos:

- Paquetes: permiten dividir un modelo en partes manejables mediante la agrupación de elementos que pueden ser casos de uso, clases o componentes. Pueden anidar otros paquetes dentro de sí.
- Dependencias: Indican que un elemento de un paquete requiere a otro de otro paquete distinto.

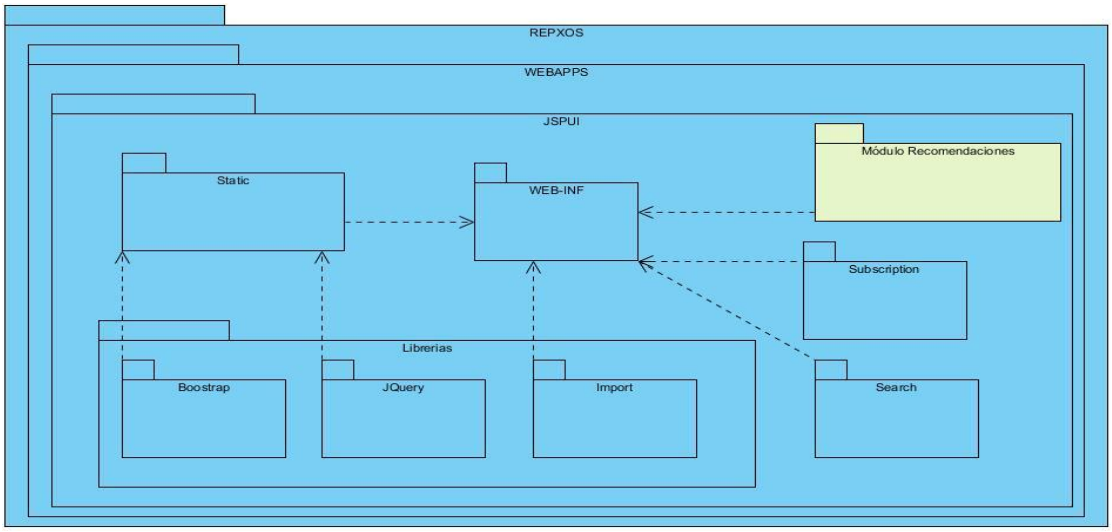

Figura 3. Diagrama de Paquetes del Módulo Recomendaciones del sistema REPXOS 3.0.

A continuación se describen los paquetes por los que está compuesto el diagrama de paquetes antes visto:

Módulo Recomendaciones: es el paquete que contiene todas las clases y las funcionalidades implementadas para su correcto funcionamiento. Se destaca en color amarillo porque contiene todas las clases que se implementan para el desarrollo del módulo

Search: es el paquete que tiene todas las clases relacionadas con el proceso de búsqueda en el sistema.

Static: es el paquete que recoge todas las librerías que se usan en el sistema y son usadas también en el Módulo Recomendaciones.

Subscription: es el paquete que contiene las vistas que permiten mostrar las listas de las colecciones y la lista de autores creados en el sistema.

WEB-INF: es el paquete principal que contiene los elementos de configuración que permiten ejecutar las operaciones en sistema, garantiza la seguridad y los roles de usuarios.

3.1.2. Diagramas de clases del diseño utilizando estereotipos web

Los diagramas de clases del diseño especifican la estructura de clases de un sistema, así como sus relaciones. Definen de forma correcta las relaciones de dependencia, generalización y asociación de clases que constituyen el sistema.

A continuación se muestra el diagrama de clases del diseño del Módulo Recomendaciones utilizando estereotipos web Ponderar Material:

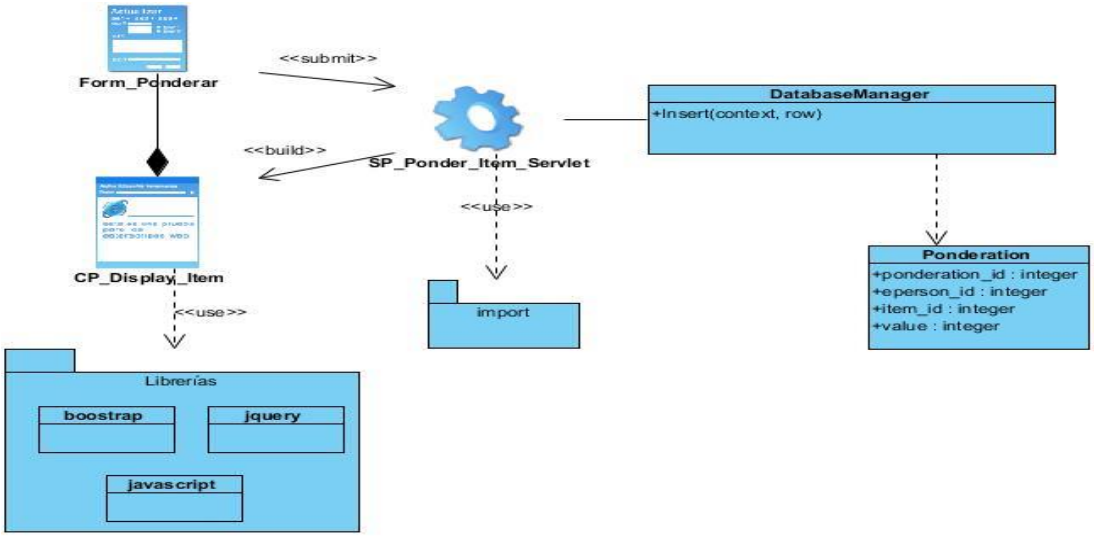

Figura 4. Diagrama de Clases del Diseño Ponderar Material

Descripción de los componentes de los diagramas de clases del diseño

En los diagramas anteriores se muestra la relación que existe entre las clases que constituyen el Módulo Recomendaciones. Los CP (*Client Page*) correspondientes a cada caso de uso, utilizan para su diseño, las librerías que se encuentran agrupadas en el paquete Librería. Con los formularios se recogen los datos que son introducidos por el usuario para luego ser procesados y posteriormente mostrar respuestas a través de los CP (*Client Page*). El paquete Import es usado por los SP *(Server Page)* del módulo para importar las clases de las que se necesiten hacer uso, creando objetos que permitan llamar aquellas funcionalidades que se encuentran definidas dentro de estas clases. Los SP *(Server Page)* son las clases creadas para asumir su correspondiente responsabilidad. DatabaseManager representa la clase de acceso a datos y emplea los métodos necesarios para poder acceder a las clases entidades que representan cada tabla de la base de datos.

3.1.3. Diagramas de Interacción

Los diagramas de interacción explican gráficamente como los objetos interactúan a través de mensajes para realizar las tareas. El UML define dos tipos de estos diagramas, ambos sirven para expresar interacciones semejantes o idénticas de mensaje: diagrama de colaboración y de secuencia.

- Diagrama de Colaboración: describen las interacciones entre los objetos en un formato de grafo o red.
- Diagrama de Secuencia: describe las interacciones en una especie de formato de cerca o muro.

Los diagramas que serán empleados en el trabajo son los de Colaboración. Estos muestra las interacciones entre objetos organizadas entorno a los objetos y los enlaces entre ellos. Destaca la organización de los objetos que participan en una interacción. Se construye colocando en primer lugar los objetos que participan en la colaboración como nodos del grafo. A continuación se representan los enlaces que conectan esos objetos como arcos del grafo. Por último, estos enlaces se adornan con los mensajes que envían y reciben los objetos. El Diagrama de Colaboración ofrece una mejor visión del escenario cuando el analista está intentando comprender la participación de un objeto en el sistema. (44)

A continuación se muestran el diagrama de colaboración asociado al requisito funcional Ponderar Material definido en el Módulo Recomendaciones:

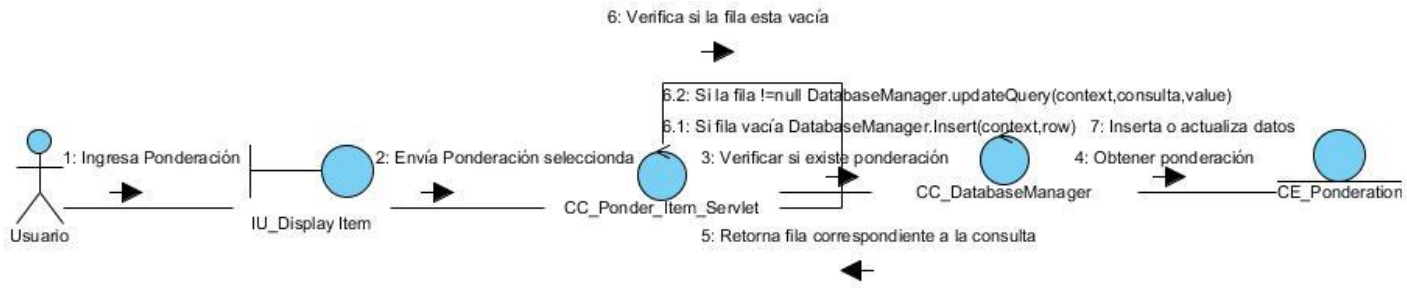

Figura 5. Diagrama de Colaboración Ponderar Material

3.2. Modelo de datos

Modelo: conjunto de conceptos que permite construir una representación organizacional de una institución.

Universo de discurso: visión del mundo real del diseñador de la Base de Datos. Su definición, constituye el primer paso del diseño para poder conseguir los objetivos del diseñador.

Modelo de Datos: conjunto de conceptos, reglas y convenciones que permiten describir, a distintos niveles de abstracción, los datos del Universo de Discurso o la estructura de una base de datos, la cual es denominada esquema, o con otras palabras, que permiten construir una representación organizada de un sistema real. Por lo tanto, se considera un modelo de datos como una herramienta que facilita la interpretación del Universo de Discurso y la representación de un sistema de información. (46)

A continuación se muestra en la **figura 6** el modelo de datos físico del Módulo Recomendaciones, el cual ofrece una descripción abstracta sobre la representación de los datos en un Sistema Gestor de Base de Datos (SGBD). Los componentes que lo conforman, son entidades existentes en el sistema REPXOS 3.0. Este modelo contiene a su vez, características propias de estos objetos y la forma en que se relacionan entre sí.

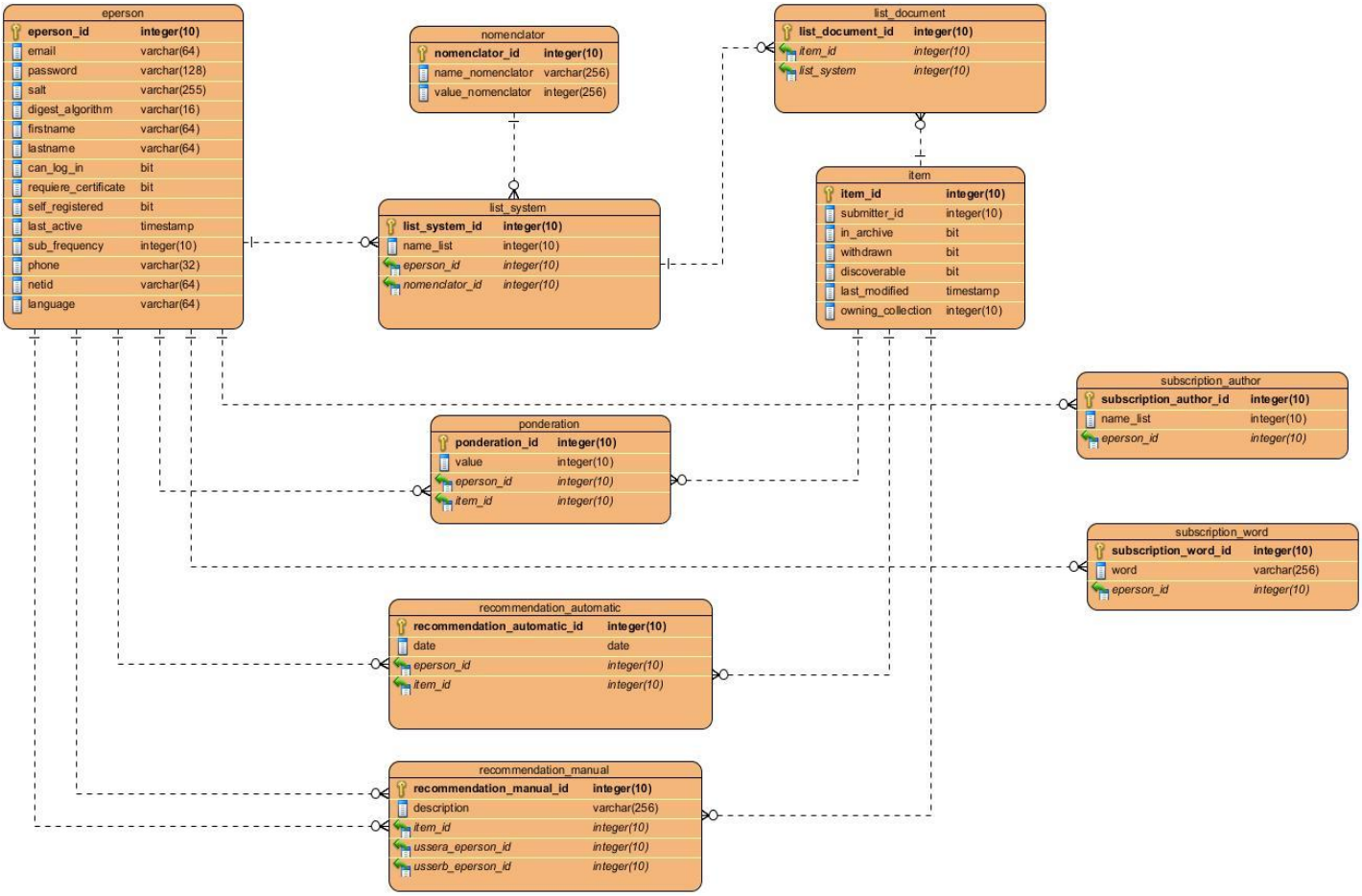

Figura 6. Diseño de la Base de Datos del Módulo de Recomendaciones

Descripción de las tablas del modelo de datos

Eperson: almacena la información correspondiente a los usuarios del sistema.

Ítem: almacena los ítems contenidos en el sistema.

List_document: almacena los documentos que han sido consultados por el usuario.

List_system: almacena las listas que han sido creadas por los usuarios en el sistema.

Nomenclator: indica de qué tipo son las listas que fue creadas en el sistema, si son públicas o privadas.

Ponderation: almacena las ponderaciones realizadas por el usuario a un ítem determinado.

Recommendation_automatic: almacena las recomendaciones realizadas por el sistema.

Recommendation_manual: almacena las recomendaciones realizadas por los usuarios del sistema.

Subscription_author: almacena el nombre de los autores a los que están suscritos los usuarios del sistema.

Subscription_word: almacena las palabras claves a los que están suscritos los usuarios del sistema.

3.3. Patrón arquitectónico

Un patrón arquitectónico se puede considerar como una descripción abstracta estilizada de buena práctica, que se ensayó y puso a prueba en diferentes sistemas y entornos. Debe describir una organización de sistema que ha tenido éxito en sistemas previos. Debe incluir información sobre cuándo es y cuándo no es adecuado usar dicho patrón, así como sobre las fortalezas y debilidades del patrón. (44)

Arquitectura en capas

Organiza el sistema en capas con funcionalidad relacionada con cada capa. Una capa da servicios a la capa de encima, de modo que las capas de nivel inferior representan servicios núcleo que es probable se utilicen a lo largo de todo el sistema.

La herramienta Dspace está constituida sobre una arquitectura en capas con 3 capas, formada por módulos y componentes independientes, lo que favorece un encapsulamiento de la funcionalidad. Debido a que REPXOS 3.0 utiliza como sistema base Dspace el patrón arquitectónico utilizado es Arquitectura en capas con 3 capas. Estas capas son: Lógica de negocio, Aplicación y Almacenamiento.

- ✓ **Capa de Lógica de Negocios:** negocia la gestión de los contenidos de los archivos junto con sus usuarios, autorizaciones y flujos de trabajos. Dentro esta capa en el sistema se implementa los

servlets, que son clases encargadas que contienen la lógica de negocio de la aplicación. En estos se implementan las funcionalidades del sistema. Entre los servlets implementados en el módulo se encuentra Ponder_Item_Servlets.class que es la clase que se encarga de controlar las evaluaciones que un usuario emite sobre el ítem que consulte.

- ✓ **Capa de Aplicación:** es la que permite la comunicación entre los distintos componentes de la aplicación. Permite la interacción entre el usuario y el sistema, para ello usan la API (*Application Programming Interface*, "Interfaz de programación de aplicaciones") pública de Dspace. En esta capa se encuentran almacenadas las interfaces contenidas en los ficheros jsp. Para el caso de este módulo un ejemplo concreto es el uso del fichero Display_Item.jsp que es la vista que muestra como quedaron definidas cada una de las funcionalidades del Módulo Recomendaciones.

- ✓ **Capa de Almacenamiento:** usa una base de datos relacional para almacenar todo el contenido de la organización, los metadatos de los contenidos, la información sobre los usuarios de los archivos, las autorizaciones y las diferentes etapas de los workflows (flujos de trabajo) que se están ejecutando. En esta capa radican las clases encargadas del almacenamiento de las entidades en la base de datos. Se destaca el uso de la clase DatabaseManager, que contiene los métodos Select, Insert, Update y Delete, los cuales permiten gestionar la información contenida en la base de datos.

3.4. Patrón de diseño utilizado en el desarrollo del Módulo Recomendaciones

Un patrón de diseño es una buena práctica documentada de la solución de un problema que ha sido aplicado satisfactoriamente en múltiples entornos. Es una solución recurrente a un problema común observado o descubierto durante el estudio o construcción de numerosos software. Su principal objetivo es incrementar la calidad del software en términos de reusabilidad, mantenimiento y extensibilidad. (34)

Los patrones GRASP (General Responsabilty Assignment Software Patterns, "Patrones de Software para la Asignación General de Responsabilidad") describen los principios fundamentales de diseño de objetos para la asignación de responsabilidades. Constituyen un apoyo para la enseñanza que ayuda a entender el diseño de objeto esencial y aplica el razonamiento para el diseño de una forma sistemática, racional y explicable. Se pueden destacar 5 patrones principales que son: Bajo Acoplamiento, Alta Cohesión, Experto, Creador y Controlador. (34)

Experto: El objetivo es asignar una responsabilidad al experto en información: la clase que cuenta con la información necesaria para cumplir con la funcionalidad.

La responsabilidad de realizar una labor es de la clase que tiene los datos involucrados. En el Módulo Recomendaciones este patrón se pone en práctica cuando la responsabilidad de realizar la evaluación de un material lo tiene la clase Ponder_Item_Servlet, de esta forma esta clase obtiene los datos correspondientes y ella decide qué hacer con ellos. Así sucede para el resto de las clases que están presentes en el módulo.

Bajo Acoplamiento: El acoplamiento es una medida de la fuerza con que una clase está conectada a otras clases, con que las conoce y con que recurre a ellas. Una clase con bajo (o débil) acoplamiento no depende de muchas otras. Una clase con alto (o fuerte) acoplamiento recurre a muchas otras. Este tipo de clases no es conveniente: presentan los siguientes problemas (34):

- Los cambios de las clases afines ocasionan cambios locales.
- Son más difíciles de entender cuando están aisladas.
- Son más difíciles de reutilizar porque se requiere la presencia de otras clases de las que dependen.

El Bajo Acoplamiento soporta el diseño de clases más independientes que reducen el impacto de los cambios, son más reutilizables y acrecientan la oportunidad de una mayor productividad.

Las clases contenidas en el Módulo Recomendaciones fueron implementadas de forma independientes, esto mitiga el impacto de los cambios que se puedan realizar en una clase.

Alta Cohesión: El objetivo es asignar una responsabilidad de modo que la cohesión siga siendo alta. La cohesión es una medida de cuan relacionadas y enfocadas están todas las responsabilidades de una clase. Una clase con baja cohesión hace muchas cosas no afines o un trabajo excesivo. No conviene este tipo de clases pues presentan los siguientes problemas (34):

- son difíciles de comprender
- son difíciles de reutilizar
- son difíciles de conservar
- son delicadas: las afectan constantemente los cambios

Las clases con baja cohesión a menudo representan un alto grado de abstracción o han asumido responsabilidades que deberían haber delegado a otros objetos.

En el Módulo Recomendaciones este patrón se pone de manifiesto al definir a cada clase del módulo una función específica y única.

Conclusiones del capítulo

Con la realización de este capítulo se han llegado a las siguientes conclusiones:

- ✓ Se diseñó el módulo y se obtuvieron los artefactos que se indican en la metodología RUP.
- ✓ Se definió como patrón arquitectónico la arquitectura N capas, definiendo 3 capas en el caso del Módulo Recomendaciones.
- ✓ Los patrones de diseño de asignación de responsabilidades GRASP utilizados en el desarrollo del módulo son: Experto, Bajo Acoplamiento, Alta Cohesión. El uso de los mismos permitió entender el diseño de la aplicación.

CAPÍTULO 4. **IMPLEMENTACIÓN Y PRUEBA DEL MÓDULO RECOMENDACIONES DEL SISTEMA PARA REPOSITORIOS DIGITALES REPXOS 3.0**

En el presente capítulo se propone explicar a partir de los resultados del diseño, la implementación del Módulo Recomendaciones del sistema REPXOS 3.0 mediante los diagramas de componente asociados a cada caso de uso, el diagrama de despliegue y los estándares de codificación. Se definen las pruebas de software como elemento esencial que garantiza la calidad del módulo y revisión final del cumplimiento de los requisitos, especificaciones de diseño y codificación.

4.1. Modelo de implementación

El modelo de implementación representa como los elementos del modelo de diseño se implementan en términos de componentes. De igual forma describe cómo se organizan los componentes de acuerdo con los mecanismos de estructuración disponibles en el entorno de implementación y la relación existente entre ellos. (33)

4.1.1. Diagrama de Componente

Un componente representa un módulo físico de código o paquete, puede ser relacionado en un diagrama de componentes. Un diagrama de componentes muestra varios componentes de un sistema, describiendo sus elementos físicos y sus dependencias. (33)

A continuación se describen el diagrama de componente Ponderar Material asociado al cada caso de uso Ponderar Material del Módulo Recomendaciones:

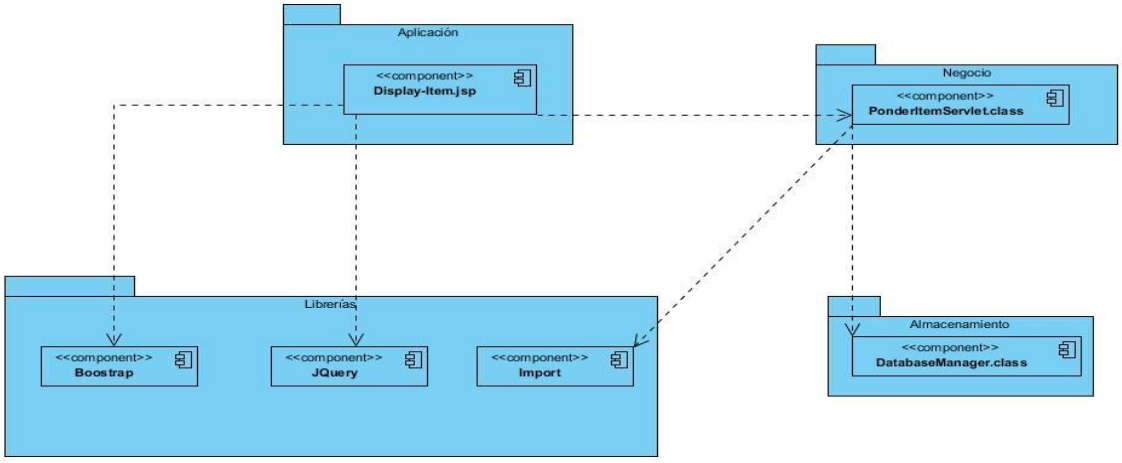

Figura 7. Diagrama de Componente Ponderar Material

4.2. Estándares de codificación

Con vistas a mejorar el entendimiento del código perteneciente al Módulo Recomendaciones para el sistema REPXOS 3.0 por otros desarrolladores y alcanzar una uniformidad en el mismo, a continuación se muestran algunos estándares de codificación utilizados en la implementación.

- Usar nombres descriptivos en inglés para las clases, propiedades y métodos.
- Iniciar el nombre de las variables de las clases con letra inicial minúscula y, en caso de ser un nombre compuesto se utiliza guion bajo (_). Ejemplos: value, ítem_id.
- Los nombres de los métodos iniciarán con minúscula y en inglés.
- Comentariar las clases y métodos de difícil comprensión.
- Los nombres de las clases deben terminar en el sufijo Servlet. Ejemplo: MyRecomendationServlet.java

4.3. Modelo de despliegue

El modelo de despliegue contiene los nodos que forman la Topología hardware sobre la que se ejecuta el Módulo Recomendaciones, el software necesario para su funcionamiento y los protocolos de comunicación. Al Módulo Recomendaciones se puede acceder desde un cliente utilizando un navegador web que cumpla con los estándares definidos por la W3C (*World Wide Web Consortium*). (33)

A continuación en la figura 8 se muestra el Diagrama de Despliegue del Módulo Recomendaciones para el Sistema REPXOS 3.0:

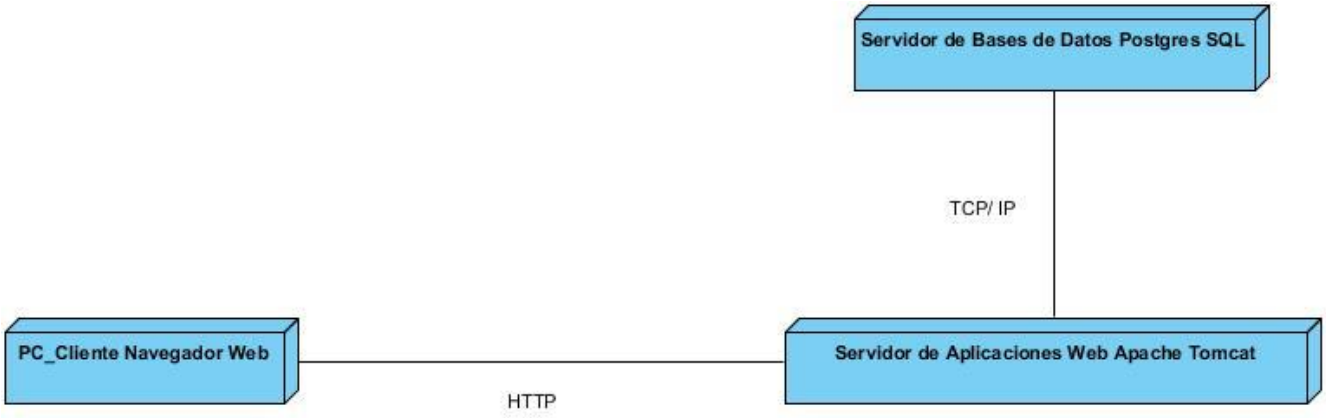

Figura 8. Diagrama de Despliegue del Módulo Recomendaciones

4.3.1. Descripción de los protocolos de comunicación

En el Diagrama de Despliegue antes visto se utilizaron los protocolos de comunicación TCP/IP y HTTP debido a que el Módulo Recomendaciones se desarrolla sobre tecnología web. Se describe a continuación cada uno de los protocolos utilizados.

TCP/IP: se utiliza para enlazar computadoras que usan sistemas operativos similares o diferentes, incluyendo pc, minicomputadoras y computadoras centrales sobre redes de área local (LAN). En el caso del sistema REPXOS 3.0 se utiliza para interconectarlo con el servidor de base de datos.

HTTP: protocolo de transferencia de hipertexto, es usado en cada transacción de la web. Define la sintaxis y semántica que utilizan los elementos software de la arquitectura web (cliente, servidor, proxy para comunicarse). En el sistema REPXOS 3.0 se utiliza este protocolo para el acceso de los clientes web a los servicios que brinda la aplicación.

4.4. Prueba

Las pruebas son una actividad en la cual un sistema o componente es ejecutado bajo unas condiciones o requerimientos especificados, los resultados son observados y registrados, y una evaluación es hecha de algún aspecto del sistema o componente.

La prueba de software es un elemento crítico para la garantía de la calidad del software y representa una revisión final de las especificaciones del diseño y de la codificación.

Elementos a tener en cuenta para que una prueba tenga éxito

- Estrategia de prueba
- Niveles de Prueba
- Tipo de Prueba
- Método de Prueba
- Caso de Prueba

Estrategia de Prueba

- Describe el enfoque y los objetivos generales de las actividades de prueba.
- Incluye los niveles de prueba a ser diseccionados, el tipo de prueba a ser ejecutada y los casos de prueba diseñados para lograr los objetivos.
- Define:

✓ Técnicas de pruebas (manual o automática) y herramientas a ser usadas.

✓ Criterios de éxitos y culminación de la pruebas.

✓ Consideraciones especiales relacionadas con los recursos necesarios para realizar las pruebas.

Una estrategia de prueba del software integra los métodos de diseño de caso de pruebas del software en una serie bien planeada de pasos que desembocará en la eficaz construcción del software. La estrategia proporciona un mapa que describe los pasos que se darán como parte de la prueba, indica cuando se planean y cuando se dan estos pasos, además de cuanto esfuerzo, tiempo y recursos consumirán. Por tanto, cualquier estrategia de prueba debe incorporar la planeación de pruebas, el diseño de casos de pruebas, la ejecución de pruebas y la recolección y evaluación de los datos resultantes.

Una estrategia de prueba del software debe ser lo suficientemente flexible como para promover un enfoque personalizado. Al mismo tiempo, debe ser lo adecuadamente rígido como para promover una planeación razonable y un seguimiento administrativo del avance del proyecto.

Niveles de Prueba

La Prueba es aplicada para diferentes tipos de objetivos, en diferentes escenarios o niveles de trabajo. Se distinguen los siguientes niveles de pruebas:

- Prueba de desarrollador
- Prueba independiente
- Prueba de unidad
- Prueba de integración
- Prueba de regresión
- Prueba de sistema
- Prueba de aceptación

Se trabajará en los siguientes niveles:

- En el nivel de unidad para desarrollarle pruebas al algoritmo propuesto mediante el método de caja blanca empleando la técnica de camino básico.
- En el nivel de sistema con el objetivo de probar las funcionalidades del Módulo Recomendaciones, esto se hará mediante el método de caja negra empleando la técnica de partición de equivalencia.
- Aplicando las pruebas de regresión para verificar que una vez que termine una iteración y se corrijan no conformidades los nuevos cambios no afecten la calidad de lo que se había hecho antes previniendo que se cometan nuevos errores.
- En el nivel de aceptación para ver si el desarrollo del módulo al sistema REPXOS 3.0 contribuyó o no a la recuperación de información de los usuarios.

Tipo de Prueba

- Funcionalidad (función, seguridad, volumen)
- Usabilidad
- Fiabilidad (integridad, estructura, stress)
- Rendimiento (benchmark, contención, carga, performance profile)
- Soportabilidad (configuración, instalación)

Los tipos de pruebas a aplicar son Funcionalidad y Rendimiento. Dentro de las pruebas de Funcionalidad se aplicarán pruebas de función. Estas pruebas fijan su atención en la validación de las funciones, métodos, servicios y casos de uso. Permiten comprobar el correcto funcionamiento de los requisitos funcionales del módulo. Mientras que dentro de las pruebas de Rendimiento se aplicarán pruebas de carga y estrés para observar el comportamiento de la aplicación. Esta prueba de carga se basa en el número esperado de usuarios

concurrentes utilizando la aplicación y la de estrés se basa en ir doblando este número de usuarios conectados para comprobar la solidez y el rendimiento de la aplicación en caso de que se supere la carga esperada. Para las pruebas de Rendimiento se utiliza la herramienta Apache JMeter en su versión 2.13. Dicha prueba arrojó como resultado que la aplicación soporta una conexión máxima de 100 usuarios concurrentemente, teniendo en cuenta que dicha prueba fue realizada en una PC con requerimientos de 4GB de RAM y un procesador Intel CORE i3 de segunda generación.

Método de Prueba: Caja Negra y de Caja Blanca

Prueba de Caja Negra

- Pruebas que se llevan a cabo sobre la interfaz del software.
- El objetivo es demostrar que las funciones del software son operativas, que las entradas se aceptan de forma adecuada y se produce un resultado correcto, y que la integridad de la información externa se mantiene (no se ve el código).

. Para desarrollar estas pruebas existen varias técnicas:

- Técnica de partición de equivalencia: esta técnica divide el campo de entrada en clases de datos que tienden a ejercitar determinadas funciones del software.
- Técnica de análisis de valores límites: esta técnica prueba la habilidad del programa para manejar datos que se encuentran en los límites aceptables.
- Técnica de Grafos de causa-efecto: es una técnica que permite al encargado de la prueba validar complejos conjuntos de acciones y condiciones.

La técnica a emplear para desarrollar las pruebas es la de partición de equivalencia. Esta técnica permite examinar los valores válidos e inválidos de las entradas existentes en el software, descubre de forma inmediata una clase de errores que, de otro modo, requerirían la ejecución de muchos casos antes de detectar el error genérico.

Prueba de Caja Blanca

- Se comprueban los caminos lógicos del software.

- Se puede examinar el estado del programa en varios puntos para determinar si el estado real coincide con el esperado (sobre el código).
- Se centran principalmente en los requisitos funcionales del software.
- Permiten encontrar: funciones incorrectas o ausentes, errores de interfaz, errores de estructuras de datos o en accesos a las bases de datos externas, errores de rendimiento y errores de inicialización y terminación.

Los métodos de prueba de caja blanca son:

- Prueba del camino básico
- Prueba de condición
- Prueba de flujo de datos
- Prueba de bucles

La técnica a emplear para desarrollar las pruebas es la del camino básico. Esta técnica permite obtener una medida de la complejidad lógica de un diseño y usar esta medida como guía para la definición de un conjunto básico. La idea es derivar casos de prueba a partir de un conjunto dado de caminos independientes por los cuales puede circular el flujo de control. Para obtener dicho conjunto de caminos independientes se construye el Grafo de Flujo asociado y se calcula su complejidad ciclomática.

Caso de Prueba

Conjunto de entradas de pruebas, condiciones de ejecución y resultados esperados desarrollados para cumplir un objetivo en particular o una función esperada. Siempre es ejecutada como una unidad, desde el comienzo hasta el final.

Debe verificar:

- Si el producto satisface los requerimientos del usuario, tal y como se describe en las especificación de los requerimientos.
- Si el producto se comporta como se desea, tal y como se describe en las especificaciones funcionales del diseño.

4.5. Prueba de Caja Negra

Con el objetivo de comprobar que las funcionalidades del Módulo Recomendaciones se realizan de forma correcta y responden a las necesidades del cliente se realizaron un total de 2 iteraciones de pruebas de caja negra, obteniéndose los siguientes resultados:

Tabla 11.Resultados de No Conformidades aplicando Prueba de Caja Negra.

No.Iteración	Aplicación			Casos de Prueba	Resueltas	No procede	Total NC
	Alta	Media	Baja				
1ra Iteración	1			12	13	0	13
2da Iteración				0	0		0

En una primera iteración se detectaron un total de 13 NC (no conformidades) divididas entre la aplicación (clasificadas en Alta, Media y Baja) y los CP (casos de pruebas). Generalmente fueron detectados errores de validación y de ortografía. En la primera iteración en la aplicación se detectó 1 NC clasificada como Alta y 12 NC detectadas en los casos de pruebas. Luego de realizar las Pruebas de Regresión con el objetivo de resolver las NC encontradas, en una segunda iteración no se detectaron NC. De manera general, las NC detectadas, no fueron significativas, por lo que no afectaban la calidad de la aplicación. Estas fueron resueltas satisfactoriamente en su totalidad.

4.6. Prueba de Caja Blanca

La técnica a emplear para desarrollar las pruebas de Caja Blanca es la del Camino Básico. Esta técnica permite obtener una medida de la complejidad lógica de un diseño y usar esta medida como guía para la definición de un conjunto básico. La idea es derivar casos de prueba a partir de un conjunto dado de caminos independientes por los cuales puede circular el flujo de control. Para obtener dicho conjunto de caminos independientes se construye el Grafo de Flujo asociado y se calcula su complejidad ciclomática.

A continuación se muestra el caso de prueba correspondiente al método **Recomendar,** el cual se localiza en la clase controladora **AuthomaticRecomendation**. El objetivo de este método es realizar las recomendaciones automáticas en el sistema a partir la lista de vecinos calculada, para el usuario activo. Se selecciona este método teniendo en cuenta la importancia que representa para el resultado de este trabajo. A continuación se muestra cómo fue aplicada la Técnica de Camino Básico al mismo.

Ver Gráfica de Flujo en el Anexo 1

Complejidad Ciclomática:

V (G)=P+1 P: nodo predicado

V (G)=23

V (G)=24

Conjunto de Rutas Independientes:

Ruta 1: 1-2-3-4-5-6-7-8-9-10-11-12-13-14-15-16-17-18

Ruta 2: 1-2-3-4-5-6-7-19-20-21-22-24-25-26-27-28-29-30-31-32-33-34-35-36-37-39-40-41-46-47-48-49-50-51-52-53-54-55-56-57-58-59-60-61

Ruta 3: 1-2-3-4-5-6-7-19-20-21-22-24-25-26-27-28-29-30-31-32-33-34-42-43-44-45-46-47-48-49-50-51-52-53-54-55-56-57-58-59-60-61

A continuación se definen los casos de prueba para cada uno de los caminos básicos obtenidos y se muestra el resultado de las pruebas aplicadas a estos caminos.

Tabla 12. Caso de prueba del camino 1

Entrada	Resultado Esperado	Objetivo
Recibe como parámetro el context, el ítem evaluado por ese usuario y luego llama a la Lista de Vecinos, del usuario activo, ordenados por peso.	Lista de recomendados.	Se garantiza la condición de que la lista de vecinos para este usuario este vacía y que al menos la lista de palabras claves y autores, para este usuario, contenga elementos.

Tabla 13. Caso de prueba del camino 2

Entrada	Resultado Esperado	Objetivo

Recibe como parámetro el context, el usuario activo y el ítem evaluado por ese usuario, luego llama a la Lista de Vecinos, del usuario activo, ordenados por peso.	Lista de recomendaciones final.	Se garantiza la condición de que existan varias ocurrencias de un mismo ítem con diferentes ponderaciones.

Tabla 14. Caso de prueba del camino 3

Entrada	Resultado Esperado	Objetivo
Recibe como parámetro el context, el usuario activo y el ítem evaluado por ese usuario, luego llama a la Lista de Vecinos, del usuario activo, ordenados por peso.	Lista de recomendaciones final.	Se garantiza la condición de que un ítem determinado no tiene varias ponderaciones, es decir aparece una sola vez con un solo voto asociado.

Conclusiones del capítulo

Con la realización de este capítulo se han llegado a las siguientes conclusiones:

- Se definieron estándares de codificación con vistas a mejorar el entendimiento del código perteneciente al Módulo Recomendaciones para el sistema REPXOS 3.0.
- Se detalló la estrategia a seguir para desarrollarle las pruebas al Módulo Recomendaciones y comprobar el correcto funcionamiento del mismo.
- Se realizaron pruebas de funcionalidad mediante los métodos de caja negra y caja blanca. En las primeras iteraciones se detectaron varias no conformidades las cuales no fueron significativas, demostrando la calidad de solución que se propone y el cumplimiento con los requisitos funcionales definidos.

CONCLUSIONES

La investigación desarrollada y los resultados obtenidos permiten a los autores arribar a las siguientes conclusiones finales:

- Se identificó que el esquema usuario-usuario del enfoque basado en memoria perteneciente al filtrado colaborativo arroja resultados ajustados a los requerimientos del usuario.
- Se implementó el Módulo Recomendaciones contribuyendo a la recuperación de los ítems más relevantes para los usuarios del sistema REPXOS 3.0 según sus preferencias.
- Se realizaron pruebas a las funcionalidades del Módulo Recomendaciones verificando la calidad de la solución que se propuso.

RECOMENDACIONES

A partir del estudio realizado en la presente investigación y teniendo en cuenta las experiencias adquiridas a lo largo de su desarrollo, los autores realizan las siguientes recomendaciones:

- Incorporar al sistema el envío de notificaciones vía correo electrónico para que los usuarios reciban recomendaciones hechas por el sistema o por otros usuarios.
- Implementar un sistema un sistema de recomendación híbrido con el objetivo de mejorar las recomendaciones realizadas en el sistema REPXOS 3.0.

REFERENCIAS BIBLIOGRÁFICAS

1. MARTÍN BUCHILLÓN, Yenni and ZAMORA CESÉ, Yanisel. SISTEMAS DE RECOMENDACIÓN. En: *Uciencia*. Universidad de las Ciencias Informáticas, 2012. p. 943.

2. MARTÍNEZ MIMBRERA, Francisco Jesús. *Sistema de Recomendación Actualizable y con Gestión de usuarios* [En línea]. Proyecto Fin de carrera. Jaén: Universidad de Jaén, 2008. Disponible: http://sinbad2.ujaen.es/cod/archivosPublicos/pfc/pfc_francisco_martinez.pdf.

3. GRCAR, M, FORTUNA, B, MLADENIC, D and GROBELNIK, M. k-NN versus SVM in the collaborative filtering framework. *Data Sciense and Classification*. 2006. P. 251–260.

4. RAMÍREZ VARGAS, M. *Servicios de recomendación en la biblioteca digital de la UDLA-P*. Licenciatura. Universidad de las Américas-Puebla, 2001.

5. LINDEN, Greg, SMITH, Brent and YORK, Jeremy. *Amazon.com Recommendations. Item-to-Item Collaborative Filtering*. Industry Report. Amazon.com, 2003.

6. TEXIER, Jose, DE GIUSTI, Marisa and OVIEDO, Nestor. *El Uso de Repositorios y su Importancia para la Educación en Ingeniería*. 2010.

7. GASTAMINZA, Félix del Valle. Documentos digitales: hipertexto y edición digital. *Universidad de Complutense Madrid*. 2006.

8. SANTIAGO. *GUÍA PARA LA CREACIÓN DE METADATOS USANDO DUBLIN CORE*. 2009. Comité de Metadatos de la Biblioteca Nacional de Chile.

9. CASTRO GALLARDO, Jorge. *Un nuevo modelo ponderado para Sistemas de Recomendación Basados en Contenido con medidas de contingencia y entropía*. Iniciación a la Investigación. Jaén: UNIVERSIDAD DE JAEN, 2012.

10. RESNICK, Paul, IACOVOU, Neophytos and SUCHAK, Mitesh. *GroupLens: An Open Architecture for Collaborative Filtering of Netnews*.

11. RESNICK, Paul and VARIAN, Hal R. Recommender Systems. *Comunications of the ACM*. 1997. Vol. 40, no. 3, p. 1–3.

12. HOUSMAN, E.M. and KASKELA, E.D. State of the art in selective dissemination of information. *IEEE Transactions on Engineering Writing and Speech*. 1970. Vol. 13, p. 78–83.

13. SEGUIDO FONT, Miguel. *Sistemas de recomendación para webs de información sobre la salud*. TESI DE MÀSTER. Universidad Politecnica de Catalunya, 2009.

14. ALLEN, R.B. User models: theory, method, and practice. *International Journal of Man-Machine Studies*. 1990. Vol. 32, p. 511–543.

15. ALBÍN RODRÍGUEZ, Antonio Pedro. *SISTEMA DE RECOMENDACIÓN COLABORATIVO BASADO EN ALGORITMOS DE FILTRADO MEJORADOS*. Proyecto Fin de carrera. Escuela Politécnica Superior de Jaén: Universidad de Jaén, 2009.

16. RICCI, Francesco, ROKACH, Lior, SHAPIRA, Bracha and KANTOR, Paul B. *Recommender Systems Handbook*. New York: Springer, 2011. ISBN 978-0-387-85819-7.

17. BALABANOVIC, M and SHOHAM, Y. Content-based, Collaborative Recommendation. *Comunications of the ACM*. 1997. Vol. 40, no. 3, p. 66–72.

18. MLADENIC, D. Text-Learning and Related Intelligent Agents: A Survey. *IEEE Intelligent Systems*. 1999. Vol. 14, no. 4, p. 44–54.

19. MITCHELL, Tom M. *Machine Learning*. New York: McGraw-Hill Sciense, 1997. ISBN 0070428077.

20. HERLOCKER, L, KONSTAN, J.A, TERVEEN, L.G and RIEDL, John. Evaluating Collaborative Filtering Recommender Systems. *ACM Transactions on Informations Systems*. 2004. Vol. 22, no. 1, p. 5–53.

21. BREESE, J.S, HECKERMAN, D and KADIE, C. Empirical analysis of predictive algorithms for collaborative filtering. En: *Proc. of the 14th Annual Conf. on Uncertainty in Artificial Intelligence*. Morgan Kaufmann, 1998. p. 43–52.

22. ADOMAVICIUS, G and TUZHILIN, A. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE Transactions on Knowledge and Data Engineering*. 2005. Vol. 17, no. 6, p. 734–749.

23. HILL, W, STEAD, L, ROSENSTEIN, M and FURNAS, G. Recommending and evaluating choices in a virtual community of use. En: *Proc. of the SIGCHI Conf. on Human Factors in Computing Systems*. New York: ACM Press/Addison-Wesley publishing Co., 1995.

24. KONSTAN, J.A, MILLER, B.N, MALTZ, D, HERLOCKER, L, GORDON, L.R and RIEDL, John. GroupLens: applyning collaborative filtering to usenet news. *Comunications of the ACM*. 1997. Vol. 40, no. 3, p. 77–87.

25. SHARDANAND, U and MAES, P. Social information filtering: Algorithms for automating “word of mouth”. En: *Proc. of the SIGCHI Conf. on Human factors in Computing Systems*. New York: ACM Press/Addison-Wesley publishing Co., 1995. p. 210–217.

26. DESHPANDE, M and KARYPIS, G. Item-based top-N recommendation algorithms. *ACM Transactions on Informations Systems*. 2004. Vol. 22, no. 1, p. 143–177.

27. HOFMANN, T. Collaborative filtering via Gaussian probabilistic latent semantic analysis. En: *Proc. of the 26th Annual Int. ACM SIGIR Conf. on Research and Development in Information Retrieval*. New York: ACM, 2003. p. 259–266.

28. Factorization meets the neighborhood: a multifaceted collaborative filtering model. En: *Proceeding of the 14th ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining*. New York: ACM, 2008. p. 426–434.

29. TACKACS, G, PILASZY, I, NEMETH, B and TIKK, D. Major components of the gravity recommendation system. *SIGKDD Exploration Newsletter*. 2007. Vol. 9, no. 2, p. 80–83.

30. FOUSS, F, RENDERS, J.M, PIROTTE, A and SAERENS, M. Random-walk computation of similarities between nodes of a graph with application to collaborative recommendation. *IEEE Transactions on Knowledge and Data Engineering*. 2007. Vol. 19, no. 3, p. 355–369.

31. GARCÍA GARCÍA, Iván. *Recuperación Optimizada en la Web Sistemas Recomendadores*. Trabajo Final. España: Universidad Politecnica de Valencia, 2014.

32. FERIA ÁLVAREZ, Liset. *Desarrollo de un sistema de recomendación colaborativo para la plataforma VideoWeb*. Trabajo de diploma para optar por el título de Ingeniero en Ciencias Informáticas. Habana: Universidad de las Ciencias Informáticas, 2012.

33. JACOBSON, Ivar, BOOCH, Grady and RUMBAUGH, James. *El Proceso Unificado De Desarrollo de Software*. Madrid: Addison Wesley, 2000. ISBN 84-7829-036-2.

34. LARMAN, Craig. *UML y Patrones Introducción al análisis y diseño orientado a objetos*. Mexico: Prentice Hall, 1999. ISBN 970-17-0261-1.

35. About Dspace. *Dspace* [En línea]. 2015. Available from: http://www.dspace.org/introducing

36. THE POSTGRESQL GLOBAL DEVELOPMENT GROUP. *PostgreSQL 9.2.1 Documentation*. WhitePaper. California: University of California, 2012.

37. MATOS, Loan Joa. *Manual de Administración de Servidores de Bases de Datos PostgreSQL con PgAdmin.*

38. GIMENO, Juan Manuel and GONZÁLEZ, José Luis. *Introducción a Netbeans*. 2011 2010.

39. THE APACHE SOFTWARE FOUNDATION. Apache Tomcat. *Apache Tomcat* [online]. 2015. Available from: http://tomcat.apache.org/index.html

40. The Apache Ant Project. *Apache Ant* [online]. 2015. Available from: http://ant.apache.org/

41. VARANASI, Balaji and BELIDA, Sudha. *Introducing Maven*. 1. New York: Apress, 2014. ISBN 978-1-4842-0841-0.

42. LÓPEZ, Ángel. Java la programación del futuro. *Compu Magazine*. 1997. Vol. 1, no. 1, p. 3–5.

43. ORACLE TEEN. *The Java Tutorials* [online]. 2014. Java Documentation. Available from: http://docs.oracle.com/javase/tutorial/

44. SOMMERVILLE, Ian. *Ingeniería de Software*. 9. México: Pearson Education, 2011. ISBN 978-607-32-0603-7.

45. HERLOCKER, Jonathan L., KONSTAN, Joseph A. and RIEDL, John. An Empirical Analysis of Design Choices in Neighborhood-Based Collaborative Filtering Algorithms. *Information Retrieval*. 2002. Vol. 5, no. 4, p. 287–310.

46. LORENZO CASTRO, Fernando. Modelo de Datos. Conceptos y clasificación. En: 2000 1999.

BIBLIOGRAFÍA

RICCI, Francesco, ROKACH, Lior, SHAPIRA, Bracha and KANTOR, Paul B. *Recommender Systems Handbook*. New York: Springer, 2011. ISBN 978-0-387-85819-

GLOSARIO DE TÉRMINOS

Algoritmo: es un conjunto ordenado y finito de operaciones sistemáticas que permiten el cálculo y hallar la solución a un tipo de problema.

Filtrado colaborativo: técnica utilizada por algunos sistemas recomendadores. Método para hacer predicciones automáticas (filtrado) sobre los intereses de un usuario.

Ítem: la unión de un documento digital con sus metadatos asociados.

Módulo: un módulo es un software que agrupa una o varias funcionalidades dentro de un sistema. Los módulos son unidades que pueden ser compiladas por separado, esto los hace reusables y permite que múltiples programadores trabajen en diferentes módulos en forma simultánea, produciendo ahorro en los tiempos de desarrollo.

Patrón arquitectónico: expresa un esquema estructural fundamental de la organización para un sistema de software, que consiste en subsistemas, sus responsabilidades e interrelaciones.

Requisitos: capacidades, condiciones o cualidades que la presente investigación debe cumplir y tener

Requisitos: capacidades, condiciones o cualidades que la presente investigación debe cumplir y tener.

RNC: Requisitos no Funcionales de Confiabilidad.

RNDI: Requisitos no Funcionales Diseño e Implementación.

RNE: Requisitos no Funcionales de Eficiencia.

RNI: Requisitos no Funcionales de Interfaz.

RNS: Requisitos no Funcionales Seguridad.

RNSE: Requisitos no Funcionales de Soporte.

RNU: Requisitos no Funcionales de Usabilidad.

Sistema de recomendación: forman parte de un tipo específico de técnica de filtro de información, los cuales presentan distintos tipos de temas o ítems de información