



Universidad de las Ciencias  
Informáticas

# **Módulo Recomendaciones de XABAL Repxos en la tecnología de Alfresco Community 5.2.**

**Trabajo de Diploma para optar por el Título de Ingeniero en Ciencias  
Informáticas**

**Autor:**

Hiram Hernández Peña

**Tutores:**

Ing. Gleidis Yuriannis Rosabal Espinosa

Ing. Gerardo Díaz Rodríguez

Ing. José Javier Hernández Benítez

**La Habana, junio del 2019**

**“Año 61 de la Revolución”**



*“Sólo la ciencia, la técnica y la productividad por hectárea podrán enfrentar el grandioso desafío que tiene por delante un planeta que se empobrece y cuya tierra agrícola y agua potable disminuyen año por año”.*

*Fidel Castro Ruz*

## **Declaración de autoría**

Declaramos ser los autores de la presente tesis y reconocemos a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

Para que así conste firmamos la presente a los \_\_\_\_ días del mes de \_\_\_\_\_ del año \_\_\_\_\_.

---

Hiram Hernández Peña

---

Ing. Gleidis Yuriannis Rosabal Espinosa

---

Ing. Gerardo Díaz Rodríguez

---

Ing. José Javier Hernández Benítez

## Agradecimientos

*Gracias de corazón a mis tutores Gleidis, Gerardo y José Javier que me ayudaron en todo momento en el desarrollo de la tesis. Gracias también por brindarme su apoyo y conocimiento, de manera incondicional.*

*Zuiero agradecer, a mi madre Ana Lucía por su apoyo y amor incondicional no solo en estos 5 años de la carrera sino en mi vida en general, siendo un referente a seguir. A mi hermana Amelia y mi cuñado Amado por ayudarme en todo momento a seguir adelante, gracias por sus consejos y fueron una guía para mi formación como profesional. A mi Pá Hiram por ser ese amigo con el que siempre se puede contar para cualquier situación. Agradezco también a mi tía Mercy por ser una segunda madre para mi que siempre ha estado siempre cuando mas lo he necesitado. Agradezco a mi primo Henry, que más que primo es mi hermano mayor. Agradezco a todas las amistades que hice durante estos 5 años en especial a Abel, Rafael, Pablo, Ibrahim y Pedro que juntos nos apoyamos y nos convertimos en una pequeña familia. También le agradezco a mi novia que se convirtió en parte importante de mi vida y que con su amor y cariño, me ayudó a perseverar en esta última etapa de mi carrera. En fin le agradezco a todos aquellos que de una forma u otra contribuyeron a convertirme en lo que soy hoy.*

## **Dedicatoria**

*Le dedico mi tesis a toda mi familia, en especial a mi mamá.*

## Resumen

El Centro de Informatización de la Gestión Documental (CIGED) de la Universidad de las Ciencias Informáticas (UCI), cuenta entre uno de sus proyectos con el Repositorio Institucional, que implementa el sistema XABAL Repxos, que gestiona las producciones científicas e investigaciones de la universidad. Este sistema cuenta con un módulo de recomendaciones que permite la creación de un espacio para compartir información entre los usuarios del mismo. Actualmente en el país, para la gestión documental no existe un proceso de recomendación que se encargue de gestionar las recomendaciones. También está la necesidad que tiene CIGED de crear de una suit de Gestión Documental. Es por ello que el presente trabajo tiene como objetivo desarrollar el Módulo Recomendaciones del sistema XABAL Repxos utilizando la tecnología de Alfresco Community 5.2 de forma que se integre al campo de la gestión documental. Para desarrollar el módulo se realizó un análisis del sistema de recomendación de XABAL Repxos y de la plataforma Alfresco. Durante el desarrollo del módulo se utilizaron como lenguajes de desarrollo JavaScript 1.6, XML 1.0, UML 2.1 y HTML5; como herramientas de desarrollo Visual Studio Code 1.23, Visual Paradigm para UML 8.0 y como metodología se utilizó AUP-UCI. Se implementó el Módulo de Recomendación utilizando las características de un sistema recomendación de filtrado colaborativo con un enfoque basado memoria y de este el esquema usuario-usuario, con las facilidades, herramientas y tecnologías de desarrollo que provee Alfresco para la integración y gestión documental.

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

## **Abstract**

The Center for Computerization of Documentary Management (CIGED) of the University of Informatics Sciences (UCI), has one of its projects with the Institutional Repository, which implements the XABAL Repxos system, which manages the scientific productions and research of the university. This system has a module of recommendations that allows the creation of a space to share information among its users. Currently in the country, for document management there is no recommendation process that is responsible for managing the recommendations. There is also the need that CIGED has to create a document management suite. That is why the present work aims to develop the Module Recommendations XABAL Repxos system using Alfresco Community 5.2 technology so that it is integrated into the field of document management. To develop the module, an analysis of the recommendation system of XABAL Repxos and the Alfresco platform was carried out. During the development of the module, JavaScript 1.6, XML 1.0, UML 2.1 and HTML5 were used as development languages; as tools of development Visual Studio Code 1.23, Visual Paradigm for UML 8.0 and as methodology AUP-UCI was used. The Recommendation Module was implemented using the characteristics of a collaborative filtering recommendation system with a memory-based approach and of this the user-user scheme, with the facilities, tools and development technologies that Alfresco provides for document integration and management.

**Key words:** memory-based approach, user-user scheme, collaborative filtering, item, recommendation systems, Alfresco platform.

## Tabla de contenido

INTRODUCCIÓN.....	1
CAPITULO 1. FUNDAMENTACIÓN TEÓRICA PARA EL DESARROLLO DEL MÓDULO RECOMENDACIONES DE XABAL REPXOS EN LA TECNOLOGÍA ALFRESCO COMMUNITY 5.2 .....	4
1.1 Conceptos asociados .....	4
1.2 Sistemas de Recomendación .....	5
1.2.1 Retroalimentación .....	5
1.2.2 Sistemas de Recomendación Colaborativos(SRC) .....	5
1.2.3 Sistemas de recomendación existentes .....	6
1.2.4 Valoración de los Sistemas de Recomendación.....	7
1.3. Metodologías de desarrollo de software .....	8
1.3.1 Metodología AUP-UCI.....	8
1.4 Entorno de desarrollo .....	10
1.4.1 Sistema Base.....	10
1.4.2 Sistema gestor de base de datos .....	11
1.4.3 Contenedor web.....	11
1.4.4 Herramienta de modelado.....	11
1.4.5 Editor de código .....	12
1.4.6 Herramienta de diseño de prototipos .....	12
1.4.7 Motor de plantillas .....	12
1.5 Lenguajes de desarrollo .....	13
1.5.1 JavaScript 1.6: .....	13
1.5.2 XML 1.0: .....	13
1.5.3 HTML 5.0: .....	13
1.5.4 UML 2.1: .....	13
Conclusiones del capítulo.....	14

CAPÍTULO 2. PROPUESTA DE SOLUCIÓN DEL MÓDULO RECOMENDACIONES DE XABAL REPXOS EN LA TECNOLOGÍA DE ALFRESCO COMMUNITY 5.2.....	15
2.1 Modelo de dominio .....	15
2.2 Propuesta de solución .....	16
2.2.1 Etapas de un sistema de recomendación basado en memoria .....	16
2.3 Especificación de requisitos .....	20
2.3.1 Requisitos funcionales .....	20
2.3.2 Requisitos no funcionales .....	22
2.4. Definición de los actores.....	24
2.5. Definición de los Casos de Uso del Módulo Recomendaciones.....	24
2.5.1. Diagrama de Caso de Uso del Módulo Recomendaciones.....	25
2.5.2 Matrices de trazabilidad .....	25
2.5.2 Descripción de los casos de usos del sistema .....	27
2.4 Arquitectura de software.....	30
2.4.1 Arquitectura En Capas .....	31
2.6 Patrones de diseño.....	32
2.6.1 Patrones GRASP .....	32
Conclusiones de capítulo .....	34
CAPÍTULO 3. IMPLEMENTACIÓN Y PRUEBA DEL MÓDULO RECOMENDACIONES DE XABAL REPXOS EN LA TECNOLOGÍA DE ALFRESCO COMMUNITY 5.2.....	35
3.1 Implementación .....	35
3.2 Diagrama de despliegue.....	35
3.3 Despliegue de la solución.....	36
3.4 Interfaces gráfica del módulo.....	37
3.5 Pruebas de software.....	38
3.5.1 Estrategia de prueba.....	39
3.5.2 Niveles de prueba .....	39

3.5.3 Tipos de prueba .....	40
3.5.4 Métodos de prueba .....	40
3.5.5 Caso de prueba .....	41
3.6 Desarrollo de las pruebas internas .....	41
3.6.1 Pruebas de caja negra .....	41
3.6.2 Pruebas de caja blanca.....	41
3.7 Resultados de las pruebas internas.....	44
Conclusiones del capítulo.....	45
CONCLUSIONES GENERALES.....	46
RECOMENDACIONES .....	47
REFENRECIAS BIBLIOGRÁFICAS .....	48
BIBLIOGRAFÍA.....	52
Anexo 1: Descripción de los CUS .....	54
Anexo 2: Interfaces gráficas del Módulo Recomendaciones .....	62
Anexo 3: Entrevista.....	65

## Índice de Ilustraciones

Ilustración 1: Escenario 2 Metodología AUP-UCI .....	10
Ilustración 2: Modelo de dominio del Módulo Recomendaciones .....	15
Ilustración 3: Diagrama de Caso de Uso del Módulo Recomendaciones .....	25
Ilustración 4: Matriz de trazabilidad requisitos funcionales-requisitos funcionales .....	26
Ilustración 5: Matriz de trazabilidad modelo de dominio-requisitos funcionales .....	26
Ilustración 6: Matriz de trazabilidad casos de uso-requisitos funcionales .....	27
Ilustración 7: Matriz de trazabilidad caso de prueba-casos de uso.....	27
Ilustración 8: Descripción de la arquitectura del Módulo Recomendaciones .....	32
Ilustración 9: Diagrama de despliegue del Módulo Recomendaciones.....	35
Ilustración 10: Estructura de carpetas del Módulo Recomendaciones.....	37
Ilustración 11: Interfaz gráfica del Módulo Recomendaciones.....	38
Ilustración 12: Código de la función getTablaDeLista() .....	43
Ilustración 13: Gráfico de flujo de la función getTablaDeLista().....	43
Ilustración 14: Interfaz de Gestionar Lista de ítems.....	62
Ilustración 15: Interfaz de mostrar ítems de la lista de ítems .....	63
Ilustración 16: Interfaz de Recomendar ítems .....	63
Ilustración 17: Interfaz de mostrar recomendaciones hechas al usuario .....	64
Ilustración 18: Mostrar ítems de la recomendación hecha al usuario .....	64
Ilustración 19: Interfaz de buscar usuario a recomendar .....	65
Ilustración 20: Interfaz de Ponderar ítems.....	65

## Índice de tablas

Tabla 1: Notación de las variables presentes en las fórmulas .....	17
Tabla 2: Requisitos funcionales del Módulo Recomendaciones .....	21
Tabla 3: Requisitos no funcionales del Módulo Recomendaciones .....	22
Tabla 4: Actores del Módulo Recomendaciones .....	24
Tabla 5: Descripción del CUS Recomendar ítems de forma manual .....	28
Tabla 6: Casos de prueba de caja blanca para el camino básico 1 .....	44
Tabla 7: Resultado de las pruebas internas ejecutadas por el desarrollador .....	44
Tabla 8: Descripción del CUS Ponderar Ítems .....	54
Tabla 9: Descripción del CUS Recomendar ítems de forma automática .....	55
Tabla 10: Descripción del CUS Buscar usuario a recomendar .....	56
Tabla 11: Descripción del CUS Gestionar recomendación de ítem a usuario .....	56
Tabla 12: Descripción del CUS Recibir notificación por correo .....	58
Tabla 13: Descripción del CUS Gestionar Lista de ítems .....	59

## INTRODUCCIÓN

Actualmente, en la sociedad continuamente se recibe y crea información apoyado por las nuevas Tecnologías de Información y Comunicación(TIC). Es tan elevada la información que existe que se hace difícil discernir qué es lo que de verdad es de interés y lo que no. Lo mismo ocurre cuando se quiere adquirir un producto (comprar un producto, ver una película, leer un libro, etc...) entre tantas opciones que se pueden encontrar gracias a las nuevas tecnologías, resulta complicado clasificar la información y decidir cuál es el producto que mejor se adapta a las necesidades a los clientes.

Con la aparición de la Web y su posterior desarrollo, surgen los Sistemas de Recomendación(SR):” Un Sistema de Recomendación es un sistema inteligente que proporciona a los usuarios una serie de sugerencias personalizadas (recomendaciones) sobre un determinado tipo de elementos (ítems). Los sistemas recomendación estudian las características de cada usuario y mediante un procesamiento de datos, encuentra un subconjunto de ítems que pueden resultar de interés para el usuario”(Jarroba, 2015). Estos son capaces de reducir la sobrecarga de información de una búsqueda definida por un usuario al posibilitarle configurar las variables de búsqueda en un dominio concreto. Esta ventaja minimiza el trabajo del usuario y la complejidad que puede tener analizar, consultar y evaluar un conjunto de información no limitada a los requisitos que se especifican.

El desarrollo de los SR ha sido ampliamente estudiado, estos son utilizados en diversos campos como el comercio electrónico, la gestión documental, entre otros. El empleo de los SR en el campo de la gestión documental ha demostrado ser efectivo sobre todo cuando se cuenta con una amplia cantidad de documentos digitales. En ocasiones el volumen de información se vuelve inmanejable por el usuario propiciando que el proceso se torne engorroso y este desista de realizar las búsquedas.(Ruiz Ricardo y Vaillant Valdéz, 2015)

Para llevar a cabo la informatización del país se ha delegado la tarea a instituciones desarrolladoras de software que han nacido en el país al calor de esta explosión tecnológica. Entre las más prestigiosas se encuentra la casa de altos estudios Universidad de las Ciencias Informáticas (UCI), pilar universitario que sirve de soporte a la industria cubana de la informática. Esta está constituida por distintos centros de desarrollo donde uno de ellos es el Centro de Informatización de la Gestión Documental (CIGED). Entre sus productos se encuentra el sistema XABAL Repxos, desarrollado por el proyecto Repositorio Institucional, el cual tiene un Módulo Recomendaciones. Este módulo permite recomendar automáticamente a los usuarios de XABAL Repxos los documentos más relevantes almacenados en el repositorio, de acuerdo a sus preferencias. Además, permite que los usuarios puedan recomendar a otros usuarios materiales consultados que hayan satisfecho sus necesidades. Los usuarios del sistema podrán emitir una evaluación de los materiales consultados, así como tener sus propias listas con estos documentos.

Actualmente en el país, para la gestión documental no existe un proceso de recomendación que se encargue de gestionar las recomendaciones. Es por ello que no se puede acceder y realizar búsquedas más rápidas de materiales, los usuarios no pueden interactuar con otros usuarios sobre información relevante que consideren que es de su mismo interés. Como consecuencia esto provoca que aumente el tiempo de adquisición de la información para los usuarios. Además de lo dicho anteriormente, también está la necesidad que tiene CIGED de crear un suit de Gestión Documental, por lo que se hace necesario el desarrollo de este módulo con tecnologías que sean compatibles con otros productos desarrollados en el centro, como es el caso del XABAL Excriba. Es por ello que se decide el desarrollo de este módulo utilizando la tecnología Alfresco.

De la situación anterior se propone como **problema a resolver**: ¿Como contribuir a mejorar la gestión documental de los usuarios del sistema XABAL Repxos?

Definiendo como **objeto de estudio**: Sistemas de recomendación.

Para darle solución al problema se plantea como **objetivo general**: Desarrollar el Módulo Recomendaciones del sistema XABAL Repxos utilizando la tecnología de Alfresco Community 5.2 de forma que se integre al campo de la gestión documental.

**Campo de acción**: Sistemas de recomendación de filtrado colaborativo.

Las **preguntas científicas** que guían y orientan el desarrollo del proceso investigativo son las siguientes:

1. ¿Cuáles son los referentes teóricos y metodológicos a tener en cuenta para abordar la solución del problema planteado relacionado con el Sistema de Recomendación en el sistema de XABAL Repxos con la tecnología de Alfresco?
2. ¿Cuál es el estado de la gestión documental de los usuarios del sistema XABAL Repxos?
3. ¿Cómo desarrollar un Módulo de Recomendaciones que permita realizar recomendaciones con la tecnología de Alfresco?
4. ¿Qué resultados tendrá la aplicación del Módulo de Recomendaciones del sistema XABAL Repxos con la tecnología de Alfresco?

Con el fin dar cumplimiento al objetivo general se plantearon las siguientes **tareas de la investigación**:

1. Sistematización de los referentes teóricos y metodológicos a tener en cuenta para abordar la solución del problema planteado relacionado con el Sistema de Recomendación en el sistema de XABAL Repxos con la tecnología de Alfresco.
2. Diagnóstico del estado de la gestión documental de los usuarios del sistema XABAL Repxos.
3. Desarrollo de un Módulo de Recomendaciones que permita realizar recomendaciones con la tecnología de Alfresco.
4. Valoración de los resultados de la aplicación del Módulo de Recomendaciones del sistema XABAL Repxos con la tecnología de Alfresco.

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.
- **Sistémico-Estructural-Funcional:** Se utilizó para analizar cómo están relacionados el proceso de recomendación a través de un suit de gestión documental.

## 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 del sistema XABAL Repxos, procesos de negocio, requisitos funcionales y no funcionales.
- **Observación:** se utiliza este método para saber cómo funciona el proceso de recomendación en el proyecto de Repositorio Institucional.

## Estructura de capítulos:

El presente documento de tesis está compuesto por: Resumen, Introducción, tres Capítulos, Conclusiones, Recomendaciones, Referencias Bibliográficas, Bibliografía, Anexos y Glosario de términos, donde se abarca todo lo relacionado con la investigación realizada.

**Capítulo 1:** Fundamentación Teórica, se incluye el estado del arte del tema tratado tanto a nivel internacional como nacional. Se describen los conceptos fundamentales asociados al dominio del problema y se hace un análisis entre las soluciones existentes y la propuesta que se realiza. Además, se analizan las herramientas, tecnologías y metodologías a usar en la solución del problema planteado.

**Capítulo 2:** Propuesta de solución, se realiza una breve descripción del Modelo de dominio, así como de sus elementos más importantes. Se especifican los requerimientos funcionales y no funcionales, a partir de los cuales se representan los casos de uso del sistema y la descripción de los mismos. Además, se expone la arquitectura, se establecen los patrones de diseño

**Capítulo 3.** Implementación y prueba, en este capítulo se presenta la distribución física del sistema mediante el diagrama de despliegue y se plantea la interfaz gráfica del módulo. Además, se trata todo lo relacionado con las pruebas realizadas al módulo desarrollado, en pos de verificar la calidad y efectividad de acuerdo con las necesidades del cliente.

## **CAPITULO 1. FUNDAMENTACIÓN TEÓRICA PARA EL DESARROLLO DEL MÓDULO RECOMENDACIONES DE XABAL REPXOS EN LA TECNOLOGÍA ALFRESCO COMMUNITY 5.2**

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 son sistemas de información que reúnen, preservan, divulgan y dan acceso a la producción digital de una comunidad, contribuyendo a aumentar su visibilidad y promoviendo la divulgación de los resultados de su actividad (TEXIER, DE GIUSTI y OVIEDO, 2010). Un repositorio digital es un sistema que almacena recursos digitales. Pueden ser ponencias de eventos, conferencias, informes de investigación, presentaciones a seminarios, tesis, textos de enseñanza y otros trabajos académicos. Específicamente los repositorios institucionales almacenan, preservan, diseminan y dan acceso a la producción intelectual de los miembros de una institución (universidad, centro de investigación, etc.). Pueden contener solamente la producción intelectual/científica (artículos, tesis, disertaciones, datos, etc.) de las instituciones, o reunir también colecciones especiales (se constituyen como bibliotecas digitales), documentación administrativa, etc.(TEXIER, DE GIUSTI y OVIEDO, 2010).

#### **Documento Digital**

Es 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, sea texto, sonido, imagen fija o en movimiento, se registra en un medio.(GASTAMINZA, 2006)

#### **Ítem**

En el sistema XABAL REPXOS 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. (SANTIAGO, 2009).

### **1.2 Sistemas de Recomendación**

En los últimos años, se ha experimentado un crecimiento enorme tanto en sitios Web como en documentos Web, el uso de Internet está cada vez más extendido. Por ello, cada vez se hace más difícil gestionar la excesiva cantidad de información a la que diariamente nos enfrentamos. Este problema se agrava cuando nos disponemos a realizar una búsqueda de información a través de Internet. Los SR se han ido consolidando como potentes herramientas para ayudar a reducir la sobrecarga de información a la que nos enfrentamos en los procesos de búsqueda de información. Ayudan a filtrar los ítems de información recuperados, usando distintas técnicas para identificar aquellos ítems que mejor satisfacen las preferencias o necesidades de los usuarios. Las recomendaciones se generan a partir de las opiniones proporcionadas por otros usuarios sobre esos ítems en búsquedas previas, o bien a partir del perfil del usuario.

#### **1.2.1 Retroalimentación**

Un sistema de recomendación debe evolucionar en el tiempo en cuanto a la calidad de sus recomendaciones. Para conseguir este objetivo se utiliza la técnica retroalimentación explícita. Esta puede ser adoptada para almacenar la retroalimentación del usuario. La técnica retroalimentación explícita es cuando el sistema requiere que el usuario explícitamente pondere los ítems (RESNICK y VARIAN, 1997). 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.

#### **1.2.2 Sistemas de Recomendación Colaborativos(SRC)**

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(BREESE, HECKERMAN y KADIE, 1998) o basado en heurística(ADOMAVICIUS y TUZHILIN, 2005) 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(HILL et al., 1995) 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 (LINDEN, SMITH y YORK, 2003) 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.

### 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.(ALBÍN RODRÍGUEZ, 2009)

### Esquema basado en usuario

Según criterios de precisión, eficiencia y estabilidad este esquema es para sistemas con un número de usuarios menor que de ítems, si la lista de ítems está en constante.(RICCI et al., 2011)

Después de haber estudiado las características de los SRC, y teniendo en cuenta que ya existe un Módulo de Recomendaciones en el sistema para repositorios digitales XABAL Repxos que utiliza el esquema basado en usuario; se escoge el mismo enfoque para este módulo en la plataforma de Alfresco.

Después de ver los conceptos en los que se basa el sistema de recomendación de XABAL Repxos, cuyo sistema está basado en el filtrado colaborativo que es capaz 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. Esto proporciona a los usuarios, conocer ítems novedosos que nunca antes habían visto y que pueden contribuir al desarrollo de investigaciones científicas.

### **1.2.3 Sistemas de recomendación existentes**

A continuación, se describen características de algunos sistemas de recomendación que se basan en el filtrado colaborativo existentes.

- ✓ **Netflix**

Netflix se caracteriza por analizar los gustos de los consumidores de manera concienzuda para utilizarlos en su beneficio. El agudo conocimiento del consumidor de la plataforma es lo que nutre al sistema de recomendaciones, que permite que los usuarios descubran más del 80% de los contenidos que consumen en la aplicación. La primera capa del sistema de recomendaciones de Netflix analiza qué es lo que ha visto un espectador, el orden y las fechas de reproducción, los programas que ha abandonado sin terminar o los que ha marcado como favoritos, entre otros datos similares. En segundo lugar, esta información de uso se combina con otros parámetros que ayudan a comprender mejor el contenido de los programas. El personal interno o externo de la plataforma visualiza minuto a minuto todos los contenidos y etiqueta de manera pormenorizada. (Arteaga, 2017)

### ✓ **eCommerce**

Los sistemas de recomendación en eCommerce utilizan un potente algoritmo de aprendizaje automático para ofrecer los productos más relevantes para cada cliente basándose en el comportamiento, la intención de compra y las compras pasadas. Se decide dónde, cómo y qué productos mostrar. Se tiene control total sobre la vista, garantizando que los productos se presenten de forma coherente con tu marca. Además, se tiene la posibilidad de establecer reglas de negocio para evitar recomendar productos comprados anteriormente o que ya están añadidos al carrito. Se puede establecer reglas por género, marca, estacionalidad, color, novedades, precio o cualquier otra variable que sea importante para tus clientes. (Sistemas de recomendación en eCommerce, 2017)

### ✓ **Pandora**

Pandora adopta un enfoque de varios niveles para evaluar y recomendar música. Un equipo de musicólogos anota canciones basadas en el género, el ritmo y la progresión. Estos datos se transforman en un vector para comparar la similitud de las canciones. Este enfoque ayuda a promover la presentación de música de cola larga de artistas desconocidos que podría ser una buena elección para un usuario en particular (Pandora, 2018).

### ✓ **XABAL Repxos**

El sistema de recomendación de XABAL Repxos, es un sistema de filtrado colaborativo que emplea las ponderaciones del usuario activo y las ponderaciones del resto de los usuarios. Utiliza un enfoque basado en memoria y de este implementa el esquema usuario-usuario, debido a que la cantidad de ítems es mayor que la cantidad de usuarios.

#### **1.2.4 Valoración de los Sistemas de Recomendación**

Tras un estudio de realizado se puede observar que no se encontró sistemas de recomendaciones de filtrado colaborativo en aplicaciones implementadas sobre la plataforma de Alfresco, por tanto para conocer el funcionamiento del proceso de recomendación, se usa como guía los SRC expuestos en la planificación del

desarrollo del Módulo Recomendaciones. Sin embargo no se puede hacer uso de estos sistemas excepto el de XABAL Repxos, que presenta las características y mayoría de las funcionalidades que necesita el módulo a desarrollar.

A continuación se exponen las causas que dieron paso a que no se utilizaran los otros sistemas de recomendación:

- Netflix analiza y construye estadísticas sobre los gustos del usuario, basándose en los datos de los usuarios registrados. Los datos los obtiene de forma implícita mediante la interacción del usuario con el sistema. El Módulo Recomendaciones a desarrollar utiliza la retroalimentación explícita.
- eCommerce construye perfiles y estadísticas sobre compras, 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.
- El SR de Pandora utiliza enfoque de varios niveles, mientras que para el desarrollo del Módulo Recomendaciones se utiliza un solo enfoque, basado en memoria.

### 1.3. Metodologías de desarrollo de software

Las metodologías de ingeniería de software pueden considerarse como una base necesaria para la ejecución de cualquier proyecto de desarrollo de software que se considere serio, y que necesite sustentarse en algo más que la experiencia y capacidades de sus programadores y equipo.

Estas metodologías son necesarias para poder realizar un proyecto profesional, tanto para poder desarrollar efectiva y eficientemente el software, como para que sirvan de documentación y se puedan rendir cuentas de los resultados obtenidos (Maida y Pacienza, 2015).

#### 1.3.1 Metodología AUP-UCI

El Proceso Unificado Ágil de Scott Ambler o *Agile Unified Process* (AUP) en inglés es una versión simplificada del Proceso Unificado de Rational (RUP). Este describe de una manera simple y fácil de entender la forma de desarrollar aplicaciones de software de negocio usando técnicas ágiles y conceptos que aún se mantienen válidos en RUP (Rodríguez Sánchez, 2015).

Al no existir una metodología de software universal, ya que toda metodología debe ser adaptada a las características de cada proyecto (equipo de desarrollo, recursos, etc.) exigiéndose así que el proceso sea configurable, se decide utilizar la variación de la metodología AUP definido para la actividad productiva de la UCI, denominada AUP-UCI (Rodríguez Sánchez, 2015).

La metodología AUP-UCI propone tres fases Inicio, Ejecución y Cierre (Rodríguez Sánchez, 2015):

- Inicio: Durante el inicio del proyecto se llevan a cabo las actividades relacionadas con la planeación del proyecto. En esta fase se realiza un estudio inicial de la organización cliente que permite obtener

información fundamental acerca del alcance del proyecto, realizar estimaciones de tiempo, esfuerzo y costo y decidir si se ejecuta o no el proyecto.

- Ejecución: En esta fase se ejecutan las actividades requeridas para desarrollar el software, incluyendo el ajuste de los planes del proyecto considerando los requisitos y la arquitectura. Durante el desarrollo se modela el negocio, obtienen los requisitos, se elaboran la arquitectura y el diseño, se implementa y se libera el producto. Durante esta fase el producto es transferido al ambiente de los usuarios finales o entregado al cliente. Además, en la transición se capacita a los usuarios finales sobre la utilización del software.
- Cierre: En esta fase se analizan tanto los resultados del proyecto como su ejecución y se realizan las actividades formales de cierre del proyecto

Para el ciclo de vida de los proyectos, AUP-UCI propone 7 disciplinas: Modelado de negocio, Requisitos, Análisis y diseño, Implementación, Pruebas Internas, Pruebas de liberación y Pruebas de Aceptación (Rodríguez Sánchez, 2015).

A partir de que el Modelado de negocio propone tres variantes a utilizar en los proyectos, Caso de Uso de Negocio (CUN), Diagrama de Proceso de Negocio (DPN) o Modelo Conceptual (MC), y existen tres formas de encapsular los requisitos, Caso de Uso del Sistema (CUS), Historia de Usuario (HU), Descripción de Requisitos por Proceso (DRP), surgen cuatro escenarios para modelar el sistema en los proyectos manteniendo en dos de ellos el MC, quedando de la siguiente forma (Rodríguez Sánchez, 2015):

- ✓ Escenario No 1: Proyectos que modelen el negocio con CUN solo pueden modelar el sistema con CUS.
- ✓ Escenario No 2: Proyectos que modelen el negocio con MC solo pueden modelar el sistema con CUS.
- ✓ Escenario No 3: Proyectos que modelen el negocio con DPN solo pueden modelar el sistema con DRP.
- ✓ Escenario No 4: Proyectos que no modelan negocio solo pueden modelar el sistema con HU.

Según la descripción de los escenarios, para el desarrollo de este módulo el escenario 2 es que más se adapta porque:

- Se aplica a los proyectos que hayan evaluado el negocio a informatizar y como resultado obtengan que no es necesario incluir las responsabilidades de las personas que ejecutan las actividades, de esta forma modelarían exclusivamente los conceptos fundamentales del negocio. Se recomienda este escenario para proyectos donde el objetivo primario es la gestión y presentación de información.(Rodríguez Sánchez, 2015)



Ilustración 1: Escenario 2 Metodología AUP-UCI

## 1.4 Entorno de desarrollo

Las herramientas y tecnologías de desarrollo son programas informáticos que usa un programador para crear, depurar, gestionar o mantener un programa.

### 1.4.1 Sistema Base

Alfresco Community Edition 5.2: es un sistema de administración de contenidos de código fuente libre, desarrollado en Java, basado en estándares abiertos y de escala empresarial para sistemas operativos tipo Windows, Unix Solaris y algunas versiones de Linux. Está diseñado para usuarios que requieren un alto grado de modularidad y rendimiento escalable. Alfresco incluye un repositorio de contenidos, un *framework* de portal web para administrar y usar contenido estándar en portales, una interfaz CIFS (Common Internet File System por sus siglas en inglés) protocolo de red que permite compartir archivos, impresoras, etcétera, entre nodos de una red de computadoras proporcionando compatibilidad de sistemas de archivos en Windows y sistemas operativos tipo Unix, un sistema de administración de contenido web, capacidad de virtualizar aplicaciones web y sitios estáticos vía Apache Tomcat, búsquedas vía el motor Apache Solr-Lucene y flujo de trabajo en jBPM (Java Business Process Model por sus siglas en inglés) (Stratebi, 2014).

#### Web Script en Alfresco Community Edition 5.2:

Un web script es un servicio vinculado a un URI que responde a métodos HTTP como GET, POST, PUT y DELETE. Son una forma de implementar API (interfaz de programación de aplicaciones) basada en REST (transferencia de estado representacional) (Alfresco Community, 2013).

#### Presentation Web Script en Alfresco Community Edition 5.2:

Los scripts de presentación web le permiten personalizar y ampliar la interfaz de usuario web. Normalmente representan HTML y pueden incluir JavaScript alojado en el navegador. A diferencia de los scripts web de datos, los scripts web de presentación se pueden alojar en el servidor de aplicaciones de contenido de Alfresco o en un servidor de presentación separado. Cuando se alojan por separado, los scripts web de presentación en el servidor de presentación interactúan con los scripts web de datos en el servidor de aplicaciones de contenido de Alfresco utilizando la API REST del repositorio (Alfresco Community, 2013).

#### Data Web Script en Alfresco Community Edition 5.2:

Los scripts de datos web encapsulan el acceso y la modificación del contenido / datos almacenados en el repositorio de contenido; por lo tanto, son proporcionados y mostrados solo por el servidor de aplicaciones de contenido de Alfresco. Proporcionan una interfaz de servidor para que las aplicaciones cliente puedan

consultar, recuperar, actualizar y realizar procesos, generalmente utilizando formatos de solicitud y respuesta como XML y JSON.(Alfresco Community, 2013)

### **Modelo de contenido en Alfresco Community Edition 5.2:**

Un modelo de contenido describe los datos que se están guardando en el repositorio. Conformar una parte esencial y crítica, ya que sin él Alfresco sería algo más que un sistema de ficheros. Un modelo de contenido presenta la siguiente información clave (ematiz, 2013):

- Los tipos de datos fundamentales y cómo deben persistirse en base de datos. Sin un modelo de contenido, Alfresco no sabría la diferencia entre un literal y una fecha.
- Los **tipos documentales** (tipos de datos de alto nivel) como carpetas y contenidos, así como los tipos documentales personalizados (procedimiento, factura, contrato...).
- **Aspectos** como “auditable” o “clasificable” o aspectos personalizados como “Datos de factura” o “Datos de empresa”.
- Propiedades (o **metadatos**) específicos de cada tipo de contenidos.
- Ligaduras (**constraints**) de ciertas propiedades, por ejemplo, una lista restringida de datos.
- Cómo indexar el contenido para las búsquedas.
- Las relaciones (**asociaciones**) entre los tipos documentales.

#### **1.4.2 Sistema gestor de base de datos**

PostgreSQL 9.4: 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. Funciona con grandes cantidades de datos y una alta concurrencia, con muchos usuarios accediendo a la vez el sistema(THE POSTGRESQL GLOBAL DEVELOPMENT GROUP, 2012). Se utilizó con el objetivo de almacenar la información con la que trabaja el Módulo Recomendaciones.

#### **1.4.3 Contenedor web**

Apache Tomcat 7.0: 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 participativo, bajo la licencia *de Apache Software License*. Apache Tomcat es empleado en un gran número de aplicaciones como Dspace, CiteSeerX y JBoss (Apache Tomcat, 2015). Se utilizó para el desarrollo y despliegue del sistema XABAL REPXOS y Alfresco Community Edition 5.2.

#### **1.4.4 Herramienta de modelado**

Visual Paradigm para UML 8.0: es una herramienta para desarrollo de aplicaciones utilizando modelado UML ideal para Ingenieros de Software, Analistas de Sistemas y Arquitectos de Sistemas que están

interesados en construcción de sistemas a gran escala y necesitan confiabilidad y estabilidad en el desarrollo orientado a objetos (Paradigm, 2013).

Posibilita la representación gráfica de los diagramas permitiendo ver el sistema desde diferentes perspectivas como: componentes, despliegue, secuencia casos de uso, clase, actividad, estado, entre otros. Además, identifica requisitos y comunica información, se centra en cómo los componentes del sistema interactúan entre ellos, sin entrar en detalles excesivos. Además, permite ver las relaciones entre los componentes del diseño y mejora la comunicación entre los miembros del equipo usando un lenguaje gráfico. Facilita licencias especiales para fines académicos (Paradigm, 2013). Se utilizó como herramienta para modelar el diseño UML del módulo.

### **1.4.5 Editor de código**

Visual Studio Code 1.23: es un editor de código fuente desarrollado por Microsoft para Windows, Linux y macOS. Incluye soporte para depuración, control de Git integrado, resaltado de sintaxis, finalización de código inteligente, fragmentos de código y refactorización de código (Valencia, 2017). Se utilizó como herramienta para escribir todo el código del módulo.

### **1.4.6 Herramienta de diseño de prototipos**

Axure RP PRO 7.0: es uno de los estándares de la industria de software orientada a generar prototipos básicos o avanzados de forma fácil. Esta herramienta está dirigida tanto a la creación de aplicaciones web como de escritorio. El programa tiene por defecto una amplia gama de widgets<sup>1</sup> con los cuales se pueden crear interfaces de una forma rápida y sencilla. Se puede crear controles personalizados mediante la combinación de los widgets existentes y la asignación de acciones en respuesta a eventos como OnClick, Onmouseover y onMouseOut o gestos como deslizar. Se podrá crear widgets personalizados (que se podrán reutilizar) que pueden simular funcionalidades o módulos que se pueden necesitar para simular un proceso, interacción, etc. Este es uno de sus puntos fuertes, la interactividad que simula el comportamiento de las interfaces hace que sea una herramienta muy útil para el de prototipado (Axure, 2014). Se utilizó para diseñar las interfaces del módulo.

### **1.4.7 Motor de plantillas**

Apache FreeMarker v2.3.28: es una biblioteca de Java para generar resultados de texto (páginas web HTML, correos electrónicos, archivos de configuración, código fuente, etc.) basados en plantillas y datos cambiantes. Las plantillas están escritas en el lenguaje de plantillas FreeMarker (FTL), que es un lenguaje simple y especializado (no un lenguaje de programación completo como PHP). Generalmente, se usa un lenguaje de programación de propósito general (como Java) para preparar los datos (emitir consultas de

---

<sup>1</sup> Pequeñas aplicaciones que tienen como principal cometido mostrar y dar fácil acceso a algunas de las principales funciones del terminal.

base de datos, hacer cálculos de negocios). Luego, Apache FreeMarker muestra los datos preparados usando plantillas. En la plantilla, se está enfocando en cómo presentar los datos, y fuera de la plantilla, se está enfocando en qué datos presentar (Freemaker, 2014). Es una de las herramientas de Alfresco, el cual genera plantillas de extensión *ftl*.

### 1.5 Lenguajes de desarrollo

Para el desarrollo de esta investigación se emplea como lenguaje de programación JavaScript 1.6 ya sea en el lado del servidor como en lado del cliente, para las interfaces se utilizó HTML5, para realizar el modelo de contenido XML 1.0 y como lenguaje de diseño UML 2.1.

#### 1.5.1 JavaScript 1.6:

JavaScript es un lenguaje interpretado con múltiples propósitos, pero se considera hasta ahora como un complemento. No requiere ningún software de servidor por lo tanto es un lenguaje de script del lado del cliente (Gauchat, 2012). Se utilizó este lenguaje Alfresco posee una Interfaz de Programación de Aplicación (API por sus siglas en inglés) en su versión 5.2 para JavaScript que permite la modificación y creación de nodos, aspectos y propiedades del repositorio de este. A través de esta API los desarrolladores de scripts podrán encontrar nodos, recorrer jerarquías de nodos, realizar búsquedas, examinar y modificar el valor de propiedades y aspectos. También, con los scripts se pueden crear nuevos ficheros, espacios o nodos de cualquier tipo.

#### 1.5.2 XML 1.0:

Es un lenguaje que permite la organización y el etiquetado de documentos. Esto quiere decir que el XML no es un lenguaje en sí mismo, sino un sistema que permite definir lenguajes de acuerdo a las necesidades (O'Reilly Media, 2014). Se utilizó ya que Alfresco trabaja las descripciones de los webscripts y el modelo de contenido con este lenguaje.

#### 1.5.3 HTML 5.0:

Es un lenguaje de marcado que se utiliza para el desarrollo de páginas de Internet. Se trata de las siglas (HTML) que corresponde a *HyperText Markup Language*, es decir, Lenguaje de Marcas de Hipertexto, que podría ser traducido como Lenguaje de Formato de Documentos para Hipertexto (Gauchat, 2012). Se utilizó para el desarrollo de las vistas.

#### 1.5.4 UML 2.1:

Lenguaje Unificado de Modelado (UML, por sus siglas en inglés) se emplea para la construcción de Modelos, 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 (Larman, 2003). Se empleó con el objetivo de visualizar el producto del trabajo (artefactos) en esquemas o diagramas estandarizados.

### **Conclusiones del capítulo**

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

- El estudio de los principales conceptos referentes a los sistemas de recomendación permitió una mejor comprensión de la presente investigación.
- A partir de que ya existe un Módulo Recomendaciones en el sistema XABAL Repxos de tipo colaborativo que utiliza el enfoque basado en memoria para las recomendaciones se tomó en cuenta utilizar el mismo tipo y enfoque para el desarrollo de este módulo.
- Luego de un estudio de las herramientas y tecnologías que utiliza Alfresco se tomó en cuenta la importancia que tiene este para la compatibilidad con otros productos del centro.

## CAPÍTULO 2. PROPUESTA DE SOLUCIÓN DEL MÓDULO RECOMENDACIONES DE XABAL REPXOS EN LA TECNOLOGÍA DE ALFRESCO COMMUNITY 5.2

En el capítulo actual se tiene como objetivo describir la propuesta de solución del Módulo Recomendaciones de XABAL Repxos en la tecnología share de Alfresco A partir del flujo descrito se realiza el Modelo de Dominio correspondiente, luego se extraen las características del Módulo de Recomendaciones a desarrollar. Se definen los requisitos funcionales y no funcionales, así como los actores y los casos de uso del sistema, exponiendo un resumen breve 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.(Larman, 2003)

Al identificar los elementos conceptuales, se decide realizar la representación de estos a través de un modelo de dominio pretendiendo lograr una comprensión más adecuada de cómo ocurren los procesos. A continuación, se muestra el modelo que describe los principales conceptos relacionado con el sistema y a su vez las relaciones entre ellos.

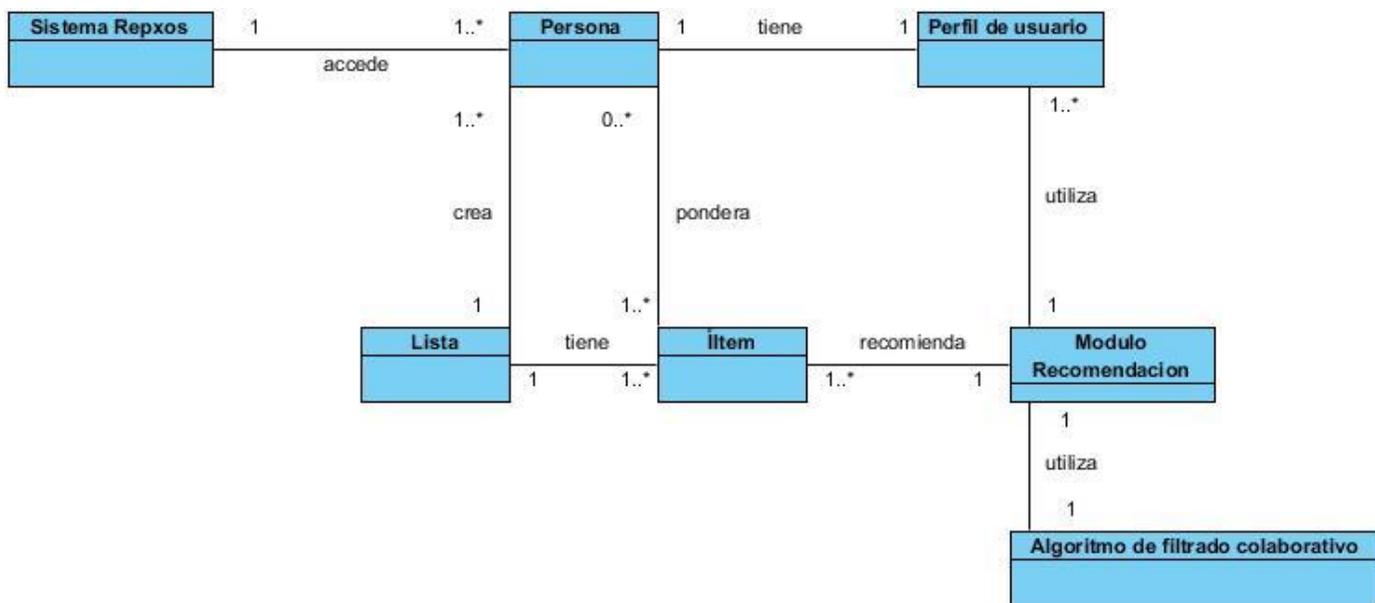


Ilustración 2: Modelo de dominio del Módulo Recomendaciones

#### Definición de los conceptos del modelo de 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 personas recomiendan.

**Lista:** constituye la lista de ítems de la persona.

**Módulo Recomendaciones:** es el encargado de mostrarle a la persona los ítems recomendados por el propio sistema o por otros usuarios.

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

**Sistema XABAL Repxos:** sistema que facilita y automatiza los procesos y servicios que brinda el repositorio a sus usuarios.

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

### 2.2 Propuesta de solución

El proceso de recomendación de documentos digitales contenidos en el sistema XABAL Repxos, permite recomendar a los usuarios del sistema materiales de acuerdo a su interés. El módulo propuesto permitirá a los usuarios del sistema XABAL Repxos, a través de la tecnología que brinda Alfresco, la posibilidad de que, una vez autenticados, puedan evaluar los ítems consultados, recomendarlos a otros usuarios, gestionar listas de ítems que pueden ser de dos tipos públicas o privadas y a su vez gestionar los ítems de dichas listas. 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. 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. Además, el módulo permitirá notificarles a los usuarios registrados cuando tienen una recomendación a través del correo electrónico.

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 y retroalimentación explícita. 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 muestran 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.

#### 2.2.1 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  $r_{ui}$  (RESNICK y VARIAN, 1997). A continuación, se presenta una tabla que contiene las notaciones de las variables.

Tabla 1: 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 $i$
$r_u$	Media de todos los votos para el usuario $u$
$Ni(u)$	$K$ usuarios más semejantes a $u$ que han ponderado $i$
$vir$	Voto dado por los KNN de $u$ para el rating $r$

### Primera etapa

La normalización de los ratings se emplea para resolver el problema que existe por las diferentes apreciaciones que los usuarios tienen a la hora de ponderar los ítems. La técnica de normalización de ratings que ha sido propuesta es Centrado por medias.

#### 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  $I_u$ .

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

**Fórmula 1.** Cálculo de la técnica de normalización Centrado por Medias (RICCI et al., 2011)

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.

En esta etapa, se decide seleccionar, 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.

Una medida popular que compara las calificaciones donde los efectos de la media y la varianza se han eliminado es la similitud de Pearson.

#### Similitud de Pearson (PC por sus siglas en inglés)

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). Esta medida indica cuan linealmente correlacionados están los elementos de un vector con los de otro.

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

**Fórmula 2.** Cálculo de la similitud de Pearson (RICCI et al., 2011)

### Configuración de la segunda etapa

La medida de similitud de Pearson se emplea para calcular la similitud entre usuarios. 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. Esta medida captura los valores negativos entre las preferencias de los usuarios y la apreciación de los diferentes ítems. El contar con estos valores negativos puede incrementar la precisión de las predicciones (HERLOCKER, KONSTAN y RIEDL, 2002). Además, ha sido demostrado que la Similitud de Pearson ha sido reconocida como la mejor medida de similitud para emplear en los sistemas de recomendación basado en memoria (HERLOCKER, KONSTAN y RIEDL, 2002).

### 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

Mejores N- Para cada usuario se calcula una lista de los n vecinos más cercanos y su respectiva similaridad es guardada.

#### 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 Clasificación.

### Clasificación

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  $i$ .

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

**Fórmula 3.** Cálculo de la predicción usando Clasificación (RICCI et al., 2011)

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:

$$\hat{r}_{ui} = h^{-1}(\arg \max_{r \in S'} \sum_{v \in N_i} \delta(r_{vi} = r) w_{uv})$$

**Fórmula 4.** Cálculo de la predicción usando Clasificación y normalizando (RICCI et al., 2011)

### Configuración de la tercera etapa

La estrategia planteada en el prefiltrado de vecinos a utilizar es:

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 y se decidió 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. 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.

### 2.3 Especificación de requisitos

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.

#### 2.3.1 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 (SOMMERVILLE, 2011).

- ✓ RF1: Ponderar ítems.
- ✓ RF2: Recomendar ítems de forma manual.
- ✓ RF3: Recomendar ítems de forma automática.
- ✓ RF4: Buscar usuario a recomendar.
- ✓ RF5: Mostrar recomendación de ítem a usuario.
- ✓ RF6: Eliminar recomendación de ítem a usuario.
- ✓ RF7: Mostrar ítems de la recomendación de ítem a usuario.
- ✓ RF8: Notificar recomendación por correo.
- ✓ RF9: Crear lista de ítems.
- ✓ RF10: Editar lista de ítems.
- ✓ RF11: Eliminar lista de ítems.
- ✓ RF12: Mostrar lista de ítems.
- ✓ RF13: Mostrar ítems de la lista de ítems.

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

Tabla 2: Requisitos funcionales del Módulo Recomendaciones

<b>Requisito funcional</b>	<b>Descripción</b>
RF1: Ponderar ítems	Se evalúa uno o varios ítems, según la apreciación del usuario, en una escala del 1 al 5, donde 5 es la máxima calificación.
RF2: Recomendar ítems de forma manual.	Permite al usuario de forma manual recomendar ítems a otros usuarios.
RF3: Recomendar ítems de forma automática	El sistema les recomienda a los usuarios ítems según su ponderación.
RF4: Buscar usuario a recomendar	Permite buscar al usuario que se le desea recomendar un documento.
RF5: Mostrar recomendación de ítem a usuario	Muestra las recomendaciones realizadas al usuario, ya sea de forma automática por el sistema o por otro usuario.
RF6: Eliminar recomendación de ítem a usuario	Permite eliminar recomendaciones realizadas al usuario.
RF7: Mostrar ítems de la recomendación de ítem a usuario	Muestra los ítems de la recomendaciones hechas ya se de forma manual o automática.
RF8: Notificar recomendación por correo	El sistema le envía un correo al usuario notificándole que tiene una recomendación, permitiendo que el usuario sepa que tiene una recomendación sin tener que acceder al sistema.
RF9: Crear lista de ítems	Permite al usuario crear una lista con todos sus ítems, dando la ventaja de realizar una recomendación más fácil sin tener hacer una búsqueda de los ítems en el repositorio, dicha lista puede ser de tipo pública o privada para que el usuario quiera o no que otros usuarios la vean.
RF10: Editar lista de ítems	Permite al usuario cambiar los datos de su lista de ítems como el nombre o el tipo.
RF11: Eliminar lista de ítems	Permite al usuario eliminar la lista de ítems, eliminando a su vez todos sus ítems.

RF12: Mostrar lista de ítems	Muestra una en una tabla la lista de ítems creada, para así poder gestionarla.
RF13: Mostrar ítems de la lista de ítems	Muestra todos los ítems de una lista de ítems.

### 2.3.2 Requisitos no funcionales

Es importante especificar las funcionalidades que debe desempeñar un sistema, así como definir todos los atributos de calidad que permiten la calidad interna y externa. Los requisitos no funcionales se reconocen como un conjunto de características de calidad, que es necesario tener en cuenta al diseñar e implementar el módulo.

La tabla muestra los requisitos no funcionales del módulo:

Tabla 3: Requisitos no funcionales del Módulo Recomendaciones

Atributo de calidad	Sub-atributos	Descripción
Usabilidad	Comprensibilidad	<ol style="list-style-type: none"> <li>1. Por cada mensaje de error del sistema debe existir una correspondencia entre el fallo real ocurrido y el texto del mensaje mostrado.</li> </ol>
Usabilidad	Cognoscibilidad	<ol style="list-style-type: none"> <li>1. Los mensajes mostrados acerca de los errores cometidos por el usuario serán notificados y los mensajes incluirán sugerencias de posibles soluciones.</li> <li>2. El módulo muestra descativadas las acciones que no han cumplido con cierta condición en función a acciones dentro del formulario.</li> <li>3. El sistema muestra las opciones de usuario en la última columna de la tabla. Por cada acción verifica si el usuario autenticado tiene</li> </ol>

		permisos a realizarla. En caso contrario el sistema no muestra la acción correspondiente.
Usabilidad	Operabilidad	<ol style="list-style-type: none"> <li>1. El sistema pide confirmación al usuario antes de ejecutar la acción.</li> <li>2. El 100% de los campos de cada formulario se podrá recorrer en orden de izquierda a derecha.</li> <li>3. El 100% de los escenarios del módulo contendrán criterios específicos para realizar búsquedas.</li> </ol>
Confiabilidad	Madurez	<ol style="list-style-type: none"> <li>1- El 100% de los formularios, el sistema impone campos obligatorios para garantizar la integridad de la información que se introduce.</li> </ol>
Funcionamiento	Software	<ol style="list-style-type: none"> <li>1- El sistema debe integrarse con el Gestor de base de datos PostgreSQL 9.4.</li> <li>2- Debe contener como contenedor web el servidor web Apache Tomcat 7.0</li> <li>3- La estación de trabajo cliente debe tener instalado un navegador web. Se recomienda utilizar Mozilla Firefox 60.X o superior.</li> </ol>

Funcionamiento	Hardware	<p>1- 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.</p> <p>2- 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).</p>
----------------	----------	--

**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 (Larman, 2003). En el Módulo Recomendaciones solo interactúa un actor, el cual se define a continuación:

Tabla 4: Actores del Módulo Recomendaciones

Actor	Descripción
Usuario	<ul style="list-style-type: none"> <li>• Persona que interactúa con el sistema.</li> <li>• Utiliza todas las funcionalidades que brinda el Módulo Recomendaciones.</li> <li>• Consulta la información que le ha sido recomendada, ya sea de forma automática o hecha por otros usuarios.</li> <li>• Recomienda ítems a otros usuarios.</li> </ul>

**2.5. Definición de los Casos de Uso del Módulo Recomendaciones**

- ✓ **CUS 1:** Ponderar ítem.

- ✓ **CUS 2:** Recomendar ítems de forma manual.
- ✓ **CUS 3:** Recomendar ítems de forma automática.
- ✓ **CUS 4:** Buscar usuario a recomendar.
- ✓ **CUS 5:** Gestionar recomendación de ítem a usuario.
- ✓ **CUS 6:** Recibir notificación por correo electrónico.
- ✓ **CUS 7:** Gestionar lista de ítems.

### 2.5.1. Diagrama de Caso de Uso del Módulo Recomendaciones

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 (Larman, 2003).

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

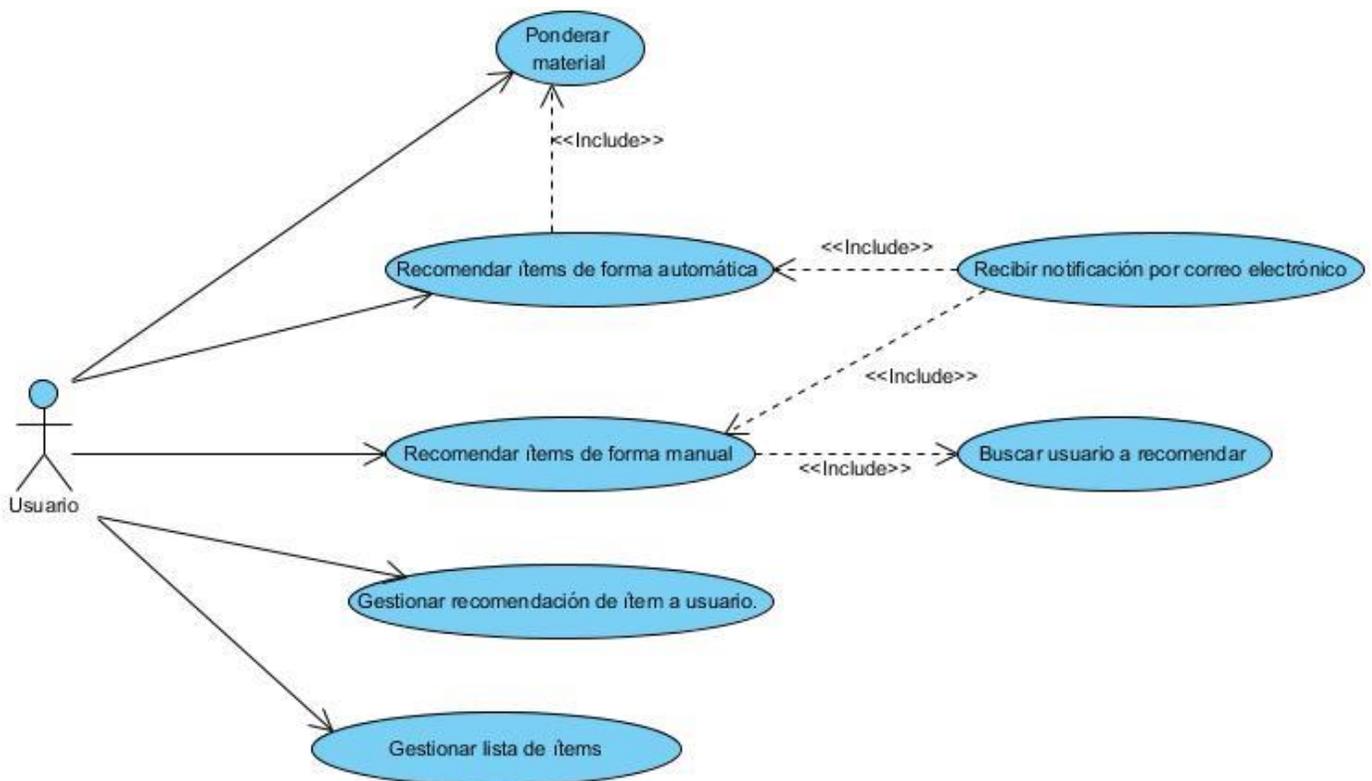


Ilustración 3: Diagrama de Caso de Uso del Módulo Recomendaciones

### 2.5.2 Matrices de trazabilidad

La matriz de trazabilidad relaciona dos elementos esenciales para la buena ejecución de las labores de un proyecto: los requisitos establecidos para dicha ejecución y el valor que cada uno de ellos agrega al conjunto

del proceso. Es una herramienta clave para la ingeniería de los proyectos, así como para el seguimiento de los diversos elementos que los componen (OBS, 2018). A continuación se presentan las matrices de trazabilidad.

(13) Requirement	Buscar usuario a rec...	Crear lista de ítems	Editar lista de ítems	Eliminar lista de ítems	Eliminar recomendaci...	Mostrar lista de ítems	Mostrar ítems de la li...	Mostrar ítems de la r...	Recibir notificación p...	Ponderar ítems	Recomendar ítems d...	Recomendar ítems d...	Mostrar recomendaci...
By: Transitor													
(13) Requirement													
Buscar usuario a recomendar													
Crear lista de ítems							✓	✓					
Editar lista de ítems							✓	✓					
Eliminar lista de ítems							✓	✓					
Eliminar recomendación de ítem a usuario									✓				✓
Mostrar lista de ítems		✓	✓	✓									
Mostrar ítems de la lista de ítems		✓	✓	✓									
Mostrar ítems de la recomendación de ítem a usu...					✓								✓
Recibir notificación por correo electrónico												✓	
Ponderar ítems											✓		
Recomendar ítems de forma automática										✓			✓
Recomendar ítems de forma manual	✓								✓				✓
Mostrar recomendación de ítem a usuario					✓			✓			✓	✓	

Ilustración 4: Matriz de trazabilidad requisitos funcionales-requisitos funcionales

(12) Requirement	Buscar usuario a recomendar	Crear lista de ítems	Editar lista de ítems	Eliminar lista de ítems	Mostrar lista de ítems	Mostrar recomendación de ítem a usu...	Mostrar ítems de la lista de ítems	Mostrar ítems de la recomendación d...	Recibir notificación por correo electró...	Ponderar ítems	Recomendar ítems de forma automática	Recomendar ítems de forma manual
By: Transitor												
(7) Class												
Algoritmo de filtrado colab...										✓	✓	
Item							✓					
Lista		✓	✓	✓	✓							
Modulo Recomendaciones	✓								✓			✓
Perfil de usuario						✓		✓				
Persona		✓								✓		
Sistema Repxos											✓	

Ilustración 5: Matriz de trazabilidad modelo de dominio-requisitos funcionales

(13) Requirement														
By: <input type="text" value="Transitor"/>		Buscar usuario a recomendar	Crear lista de ítems	Editar lista de ítems	Eliminar lista de ítems	Eliminar recomendación de...	Mostrar lista de ítems	Mostrar recomendación de...	Mostrar ítems de la lista de...	Mostrar ítems de la recom...	Recibir notificación por cor...	Ponderar ítems	Recomendar ítems de form...	Recomendar ítems de form...
(7) Use Case														
<input type="radio"/>	Buscar usuario a recomendar	✓												
<input type="radio"/>	Gestionar lista de ítems		✓	✓	✓		✓		✓					
<input type="radio"/>	Gestionar recomendación ...					✓		✓		✓				
<input type="radio"/>	Ponderar ítems											✓		
<input type="radio"/>	Recibir notificación por cor...										✓			
<input type="radio"/>	Recomendar ítems de form...												✓	
<input type="radio"/>	Recomendar ítems de form...													✓

Ilustración 6: Matriz de trazabilidad casos de uso-requisitos funcionales

(6) Use Case							
By: <input type="text" value="Transitor"/>		Buscar usuario a recomendar	Gestionar lista de ítems	Gestionar recomendación ...	Ponderar ítems	Recibir notificación por cor ...	Recomendar ítems de form...
(6) Test Case							
<input type="radio"/>	Buscar usuario a recomendar	✓					
<input type="radio"/>	Gestionar lista de ítems		✓				
<input type="radio"/>	Gestionar recomendación ...			✓			
<input type="radio"/>	Ponderar ítem				✓		
<input type="radio"/>	Recibir notificación por cor...					✓	
<input type="radio"/>	Recomendar ítems de form...						✓

Ilustración 7: Matriz de trazabilidad caso de prueba-casos de uso

## 2.5.2 Descripción de los casos de usos del sistema

### CUS 2: Recomendar ítems de forma manual.

Tabla 5: Descripción del CUS Recomendar ítems de forma manual

<b>Objetivo</b>	Recomendar ítems a otros usuarios de forma manual.	
<b>Actores</b>	Usuario	
<b>Resumen</b>	El caso de uso inicia cuando el usuario necesita recomendarle ítems a otros usuarios.	
<b>Complejidad</b>	Alta	
<b>Prioridad</b>	Alta	
<b>Precondiciones</b>	Una lista de ítems ha sido creada.	
<b>Postcondiciones</b>	Uno o varios ítems fueron recomendados.	
<b>Flujo de eventos</b>		
<b>Flujo básico Recomendar ítems</b>		
	<b>Actor</b>	<b>Sistema</b>
1.	Selecciona la opción "Recomendar ítems".	
2.		<p>Muestra un formulario con los siguientes campos:</p> <ul style="list-style-type: none"> <li>- Seleccione una lista de ítems</li> <li>- Seleccionar Ítems</li> <li>- Ítems a recomendar</li> <li>- Seleccione los usuarios a recomendar</li> <li>- Observaciones de la recomendación</li> </ul> <p>Muestra las siguientes opciones:</p> <ul style="list-style-type: none"> <li>- Añadir ítems a recomendación</li> <li>- Recomendar</li> </ul>
1.	Selecciona una lista de ítems.	
2.		Muestra los ítems de la lista seleccionada.
3.	Selecciona los ítems a recomendar.	
4.	Selecciona la opción "Añadir ítems a recomendación"	

5.		Comprueba que no se han añadidos los mismos ítems.
6.		Añade los ítems seleccionados al campo ítems a recomendar.
7.	Selecciona los usuarios a recomendar.	
8.	Da clic en la opción “Recomendar”	
9.		Comprueba que se han llenado los campos correctamente.
10.		Muestra un mensaje: “Se ha realizado una nueva recomendación”.
11.		Termina el caso de uso.
<b>Flujos alternos</b>		
<b>5a. Se han añadidos los mismos ítems.</b>		
	<b>Actor</b>	<b>Sistema</b>
1.		Comprueba que el ítem seleccionado ya este añadido.
2.		Muestra un mensaje: “Ya el ítem se encuentra en lista de recomendados”.
<b>10ª. Los datos son incorrectos</b>		
	<b>Actor</b>	<b>Sistema</b>
1.		Comprueba que se han introducidos datos incorrectos.
2.		Señala los campos que deben ser llenados, indicando que existen errores en el formulario.
<b>Relaciones</b>	<b>CU incluidos</b>	Buscar usuario a recomendar, <u>Ver CU</u> <u>Buscar usuario a recomendar</u>
	<b>CU extendidos</b>	N/A
<b>Prototipo elemental de interfaz gráfica de usuario</b>		

The image shows a web form with the following elements:

- Seleccionar Lista de Ítems:** A dropdown menu.
- Seleccionar Ítems:** A list box with a vertical scrollbar.
- Añadir ítems a recomendar:** A button located below the 'Seleccionar Ítems' list box.
- Ítems a recomendar:** A larger list box with a vertical scrollbar.
- Seleccionar Usuarios a recomendar:** A dropdown menu.
- Descripción de la recomendación:** A large text area.
- Recomendar:** A button located at the bottom right of the form.

Para más información de la descripción de los CUS consultar el [anexo 1](#).

## 2.4 Arquitectura de software

El diseño arquitectónico representa la estructura de los datos y de los componentes del programa que se requieren para construir un sistema basado en computadora. Considera el estilo de arquitectura que adoptará el sistema, la estructura y las propiedades de los componentes que lo constituyen y las interrelaciones que ocurren entre sus componentes arquitectónicos (Pressman, 2010).

Como el Módulo de Recomendaciones del sistema XABAL Repxos es en la tecnología de Alfresco que utiliza la arquitectura en Capas o N-Capas, como también es conocida, el desarrollo de este módulo utilizará dicha arquitectura.

### 2.4.1 Arquitectura En Capas

La arquitectura basada en capas se enfoca principalmente en el agrupamiento de funcionalidad relacionada dentro de una aplicación en distintas capas que son colocadas verticalmente una encima de otra, la funcionalidad dentro de cada capa se relaciona con un rol o responsabilidad específica. El dividir en capas una aplicación, permite la separación de responsabilidades lo que proporciona una mayor flexibilidad y un mejor mantenimiento (Muñoz Serafín, 2018).

En una aplicación con una capa de presentación, una capa de lógica y una capa de acceso a datos, la responsabilidad de la capa de presentación es la de interactuar con el usuario, solicitando y proporcionando la información que el usuario requiere. La responsabilidad de la capa de lógica de negocio es la de hacer cumplir las reglas de negocio o requerimientos de la aplicación mientras que la responsabilidad de la Capa de acceso a datos es la de recuperar y modificar datos del origen de datos.(Muñoz Serafín, 2018)

#### **Capa de presentación**

En esta capa se brindan dos clientes, el cliente web XABAL Repxos y el explorador de Alfresco:

- El cliente web de XABAL Repxos es un Repositorio Digital cuyo objetivo es brindar servicios que cumplan con los requisitos de la gestión documental, apoyando a las actividades de los procesos documentales.
- El explorador de Alfresco es incluido en el núcleo como una herramienta de administración de apoyo, ya que expone todas las funcionalidades del ECM utilizado y es requerido para llevar a cabo acciones más avanzadas como el mantenimiento de las reglas de negocio y la gestión de grupos y usuarios

#### **Capa de lógica de negocio**

Esta capa encapsula los servicios de datos, los cuales brindan servicios avanzados para la gestión de contenidos y son incluidos dentro del repositorio de Alfresco, llamados a través de una interfaz de servicios que coordina la conversación con los clientes del servicio o invoca cualquier flujo o componente de negocio.

#### **Capa de datos**

Esta capa agrupa al sistema de ficheros y la base de datos. El sistema de ficheros está compuesto a la vez por los ficheros del contenido y los índices de búsqueda. En esta capa se guarda toda la información y el contenido que es gestionado por el núcleo cuando realiza sus funciones (Muñoz Serafín, 2018). A continuación, se muestra la arquitectura para el Módulo Recomendaciones.

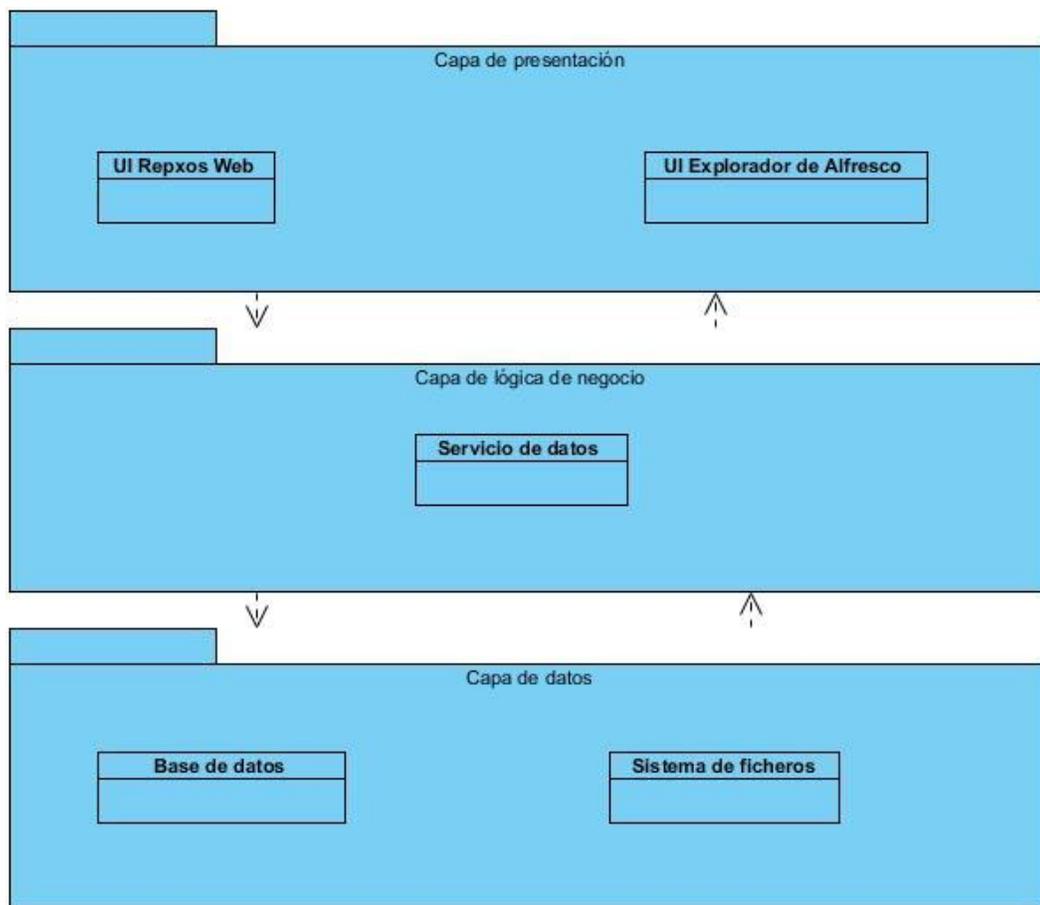


Ilustración 8: Descripción de la arquitectura del Módulo Recomendaciones

## 2.6 Patrones de diseño

Un patrón es una descripción de un problema y la solución, a la que se da un nombre, y que se puede aplicar a nuevos contextos; idealmente, proporciona consejos sobre el modo de aplicarlo en varias circunstancias, y considera los puntos fuertes y compromisos (Larman, 2003).

### 2.6.1 Patrones GRASP

Los Patrones GRASP (General Responsibility Assignment Software Patterns) describen los principios fundamentales de la asignación de responsabilidades a objetos, expresados en forma de patrones (Larman, 2003).

**Patrón Experto:** Es la clase que tiene la información necesaria para realizar la responsabilidad. Este se utiliza a la hora de modelar las entidades persistentes de la solución donde las funcionalidades propias de cada información reimplementan en la clase que contiene toda la información, o sea, la clase experta en la información (Larman, 2003).

Este patrón se puede evidenciar cuando la responsabilidad de realizar la evaluación de un material lo tiene el fichero `ponderarItems.js`, de esta forma este obtiene los datos correspondientes y decide qué hacer con ellos.

**Patrón Creador:** guía la asignación de responsabilidades relacionadas con la creación de objetos, tarea muy frecuente en los sistemas orientados a objetos. El propósito fundamental de este patrón es encontrar un creador que se debe conectar con el objeto producido en cualquier evento. Al escogerlo como creador, se da soporte al bajo acoplamiento (Larman, 2003).

Este patrón se manifiesta en la funcionalidad de adicionar listas de ítems que es la encargada de crear los nodos Lista de Ítems.

**Patrón Controlador:** propone el diseño de clases con la responsabilidad de controlar el flujo de eventos del sistema a clases específicas. Es más que asignar la responsabilidad del manejo de un mensaje de los eventos de un sistema a una clase (Larman, 2003).

Este patrón se evidencia en el fichero `action.js` que se encarga controlar muchas de las acciones que se realizan con los temas.

**Patrón Bajo Acoplamiento:** es un principio que se debe recordar durante las decisiones de diseño: es la meta principal que es preciso tener presente siempre. Es un patrón evaluativo que el diseñador aplica al juzgar sus decisiones de diseño. El acoplamiento es una medida de la fuerza con que una clase está conectada a otras, con las que conoce y con que recurre a ellas. El patrón propone un diseño de clases más independiente, lo que reduce el impacto del cambio y facilita la reutilización en otros sistemas (Larman, 2003).

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

**Patrón 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 (Larman, 2003):

- 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.

### **Conclusiones de capítulo**

Una vez realizada la propuesta de solución de la investigación se arribó a las siguientes conclusiones:

- La definición del modelo de dominio con su respectiva descripción, facilitó el entendimiento del negocio para el desarrollo del módulo en cuestión.
- La definición de los requisitos funcionales permitió identificar las funcionalidades a desarrollar en el Módulo de Recomendación.
- La arquitectura de software empleada y los patrones de diseño sentaron las bases para la siguiente etapa de construcción del software.

## CAPÍTULO 3. IMPLEMENTACIÓN Y PRUEBA DEL MÓDULO RECOMENDACIONES DE XABAL REPXOS EN LA TECNOLOGÍA DE ALFRESCO COMMUNITY 5.2

El presente capítulo tiene como objetivo la implementación y prueba del módulo para lograr un producto de calidad. Para ello es necesario trazarse un plan de pruebas desde el principio, darles seguimiento a los cambios, controlar el funcionamiento de la aplicación y su despliegue.

### 3.1 Implementación

Según la disciplina de implementación que propone la metodología AUP-UCI, a partir de los resultados del Análisis y Diseño se construye el sistema (Rodríguez Sánchez, 2015). Por tanto, la implementación del módulo se hizo a través de webscripts de presentación, webscripts de datos y modelo de contenidos en la tecnología de Alfresco Community 5.2.

### 3.2 Diagrama de despliegue

El despliegue es el proceso de asignar artefactos a nodos o instancias de artefactos a instancias de nodos. Los diagramas de despliegue muestran:

- El hardware sobre el que se ejecutará el sistema y cómo el software se despliega en ese hardware.
- La configuración de los nodos que participan en la ejecución y de los artefactos que residen en los nodos.

En UML se utilizan para visualizar los aspectos estáticos de los nodos físicos y sus relaciones y para especificar sus detalles para la construcción (Ruiz, 2017). En la siguiente ilustración se muestra el diagrama de despliegue del Módulo Recomendaciones.

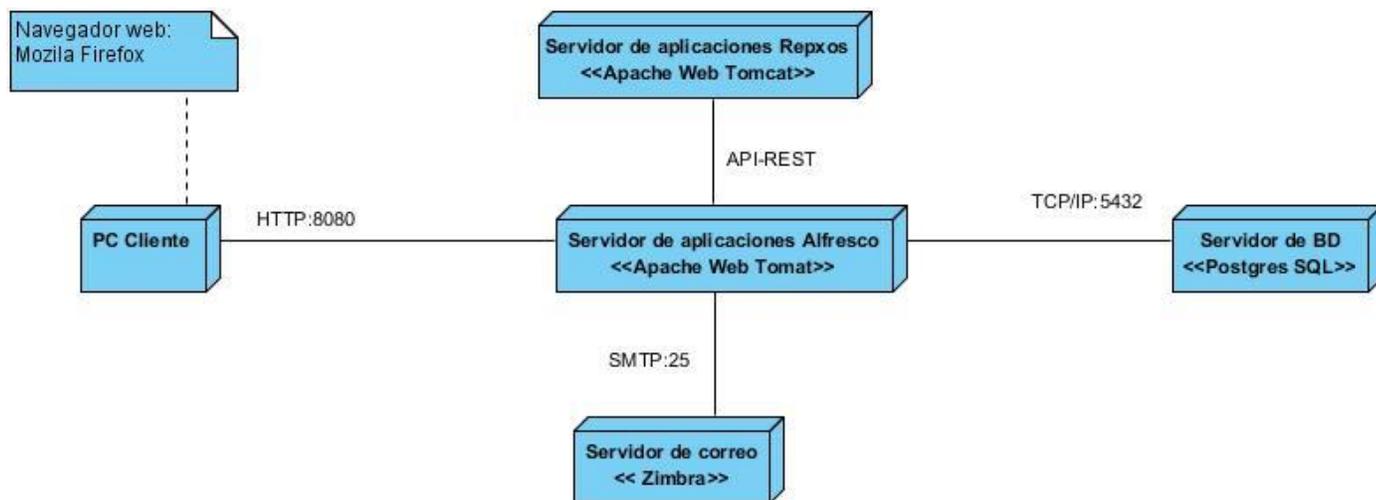


Ilustración 9: Diagrama de despliegue del Módulo Recomendaciones

A continuación, se describen los nodos:

**PC cliente:** Es el nodo que contiene los navegadores web (Firefox, Chrome, etc.) para acceder a las páginas cliente que interactúan con las páginas servidoras. Se procesan todas las interfaces de usuario que establecen la comunicación entre el usuario y el Módulo Recomendaciones.

**Servidor de aplicaciones Alfresco:** Este nodo tiene alojado Alfresco Community Edition 5.2 que es donde se construye la aplicación, hosteado en un servidor web Apache, y usa la API-REST del sistema XABAL Repxos para capturar los ítems del repositorio.

**Servidor de aplicaciones XABAL Repxos:** En este se encuentra desplegado el sistema XABAL Repxos para poder acceder a la API-REST.

**Servidor de correo:** Este nodo está desplegado el servidor de correo Zimbra.

**Servidor de base de datos:** En este servidor es donde se encuentra el sistema gestor de bases de datos PostgreSQL. Aquí se encuentra toda la información estructurada.

### Descripción de las relaciones entre los nodos

Primeramente, el usuario al interactuar con el módulo a través de la computadora cliente, este se comunica por el protocolo HTTP y el puerto 8080 al servidor de aplicaciones, donde se encuentra el servidor web apache Tomcat para Alfresco, el cual se comunica con servidor web apache Tomcat para XABAL Repxos, para utilizar la API-REST del repositorio. El servidor de aplicaciones a su vez se conecta con el servidor de base de datos por el protocolo TCP/IP puerto 5432, para acceder a la información almacenada y referente con el módulo, y el servidor de correo por el protocolo SMTP puerto 25.

### 3.3 Despliegue de la solución

A continuación, se muestra el despliegue de la solución del Módulo de Recomendación, donde se expone la estructura de carpetas y archivos con que cuenta el módulo.

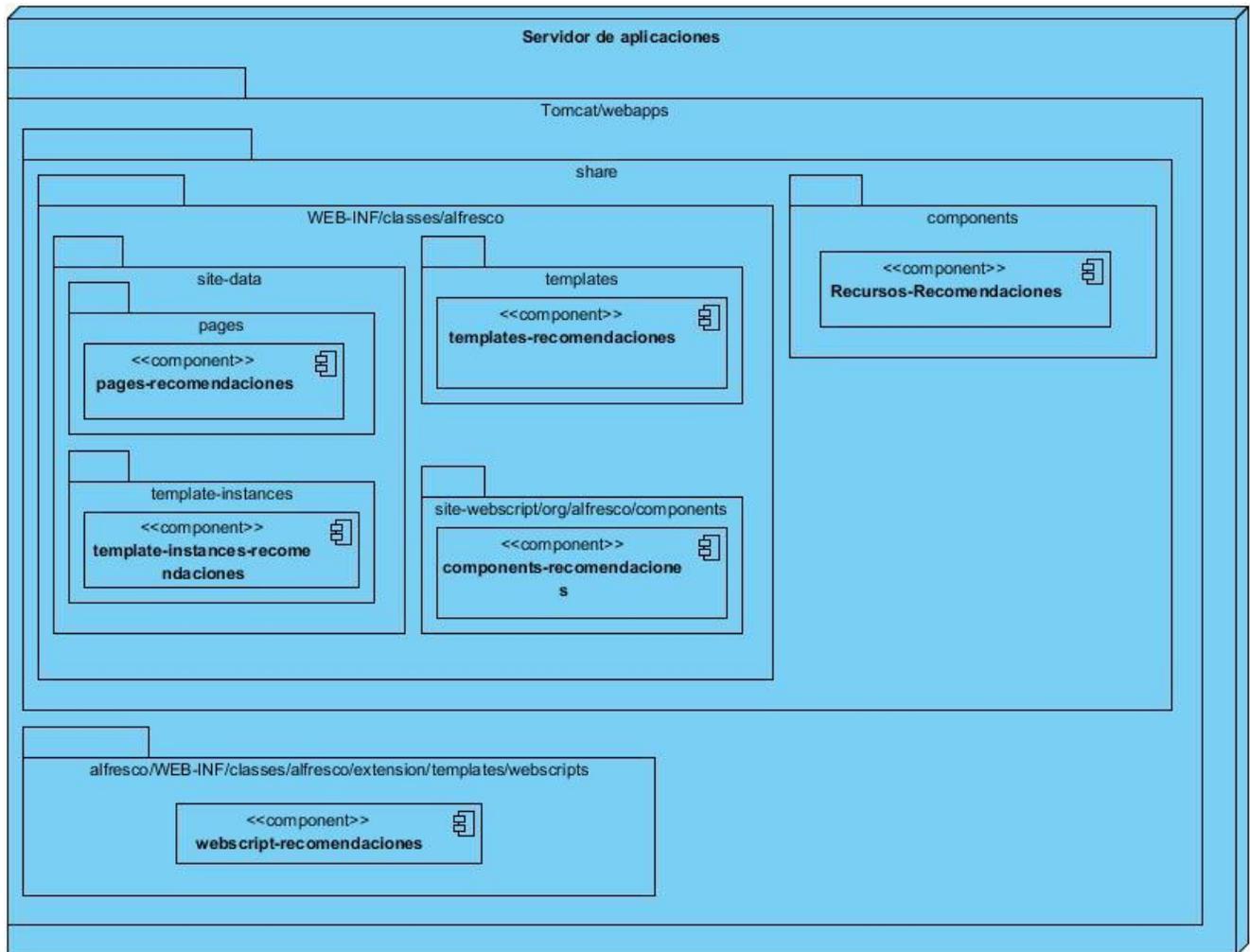


Ilustración 10: Estructura de carpetas del Módulo Recomendaciones

### 3.4 Interfaces gráfica del módulo

A continuación, se muestra como queda la interfaz principal del Módulo Recomendaciones.

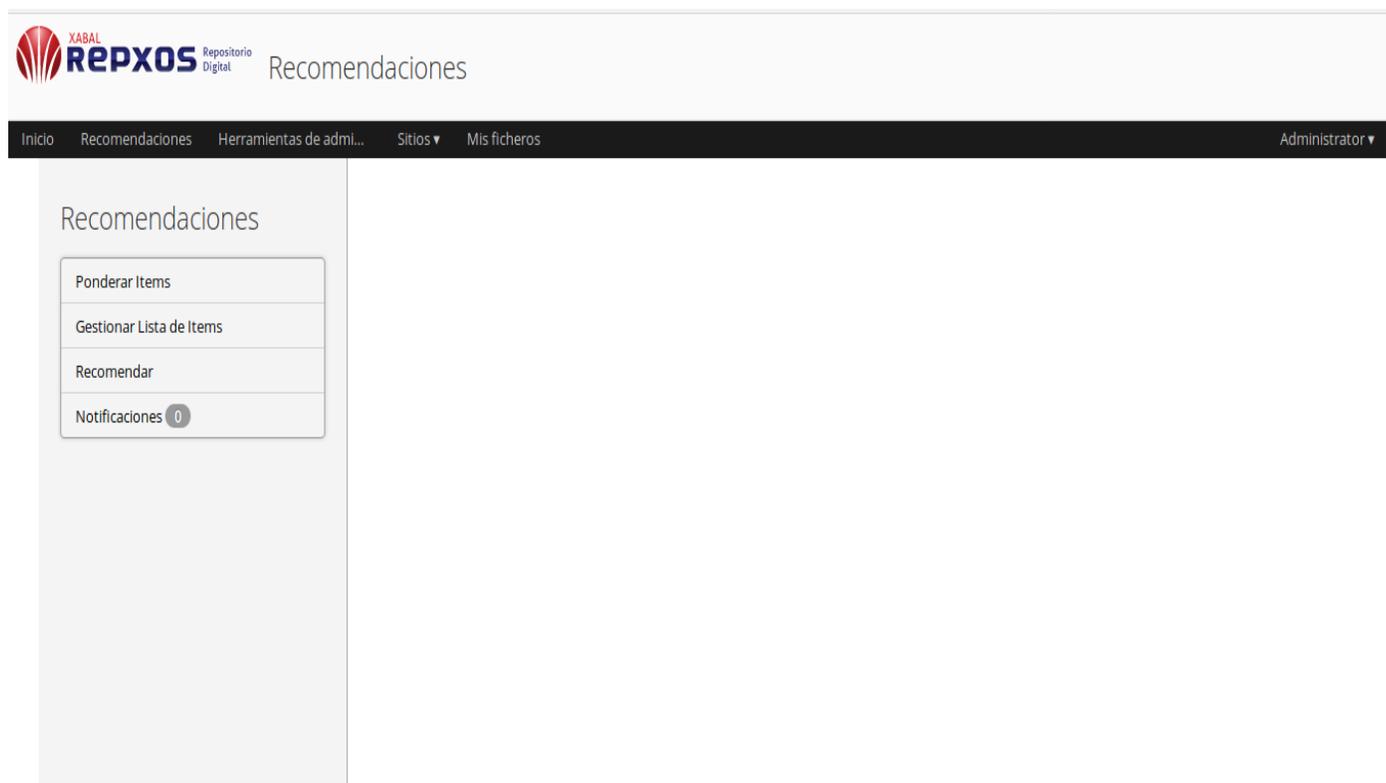


Ilustración 11: Interfaz gráfica del Módulo Recomendaciones

Para más información acerca de las interfaces del módulo consultar el [anexo 2](#).

### 3.5 Pruebas de software

El software se prueba para descubrir errores que se cometieron de manera inadvertida conforme se diseñó y construyó. La prueba es un conjunto de actividades que pueden planearse por adelantado y realizarse de manera sistemática. Por esta razón, durante el proceso de software, debe definirse una plantilla para la prueba del software: un conjunto de pasos que incluyen métodos de prueba y técnicas de diseño de casos de prueba específicos. Las pruebas representan el último bastión desde donde puede valorarse la calidad y, de manera más pragmática, descubrirse errores (Pressman, 2010).

La meta de probar es encontrar errores, y una buena prueba es aquella que tiene una alta probabilidad de encontrar uno, en sí mismas deben mostrar un conjunto de características que logren la meta de encontrar la mayor cantidad de errores con el mínimo esfuerzo (Pressman, 2010).

Elementos a tener en cuenta para que una prueba tenga éxito (Lara, 2013):

- Estrategia de prueba.
- Niveles de prueba.
- Tipo de prueba.
- Método de prueba.

- Caso de prueba.

### 3.5.1 Estrategia de prueba

Una estrategia de prueba de software proporciona una guía que describe los pasos que deben realizarse como parte de la prueba, cuándo se planean y se llevan a cabo dichos pasos, y cuánto esfuerzo, tiempo y recursos se requerirán. Por tanto, cualquier estrategia de prueba debe incorporar la planificación de la prueba, el diseño de casos de prueba, la ejecución de la prueba y la recolección y evaluación de los resultados (Pressman, 2010).

### 3.5.2 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 (Ruiz Ricardo y Vaillant Valdéz, 2015):

- 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 haciendo uso del método de caja blanca para realizar pruebas del código implementado haciendo uso de la técnica camino básico.
- En el nivel de sistema con el objetivo de probar las funcionalidades del módulo, 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.

Es válido aclarar que la metodología AUP-UCI propone 3 disciplinas de pruebas para aplicarle a los proyectos: pruebas internas, pruebas de liberación y pruebas de aceptación (Rodríguez Sánchez, 2015).

Para dicha investigación se desarrollará la disciplina Pruebas internas.

- 1- Pruebas internas: En esta disciplina se verifica el resultado de la implementación probando cada construcción, incluyendo tanto las construcciones internas como intermedias, así como las versiones finales a ser liberadas. Se deben desarrollar artefactos de prueba como: diseños de casos de prueba, listas de chequeo y de ser posible componentes de prueba ejecutables para automatizar las pruebas.

Se utilizó la técnica de la partición de equivalencia que permite examinar los valores válidos y no válidos de las entradas existentes en el software.

### 3.5.3 Tipos de prueba

Un grupo de actividades de pruebas pueden tener por objetivo verificar el sistema de software (o parte el sistema) en base a un motivo u objetivo específico para probar. Un tipo de prueba se centra en un objetivo de prueba en particular, que puede ser cualquiera de los siguientes (ISTQB, 2011):

- Una función a realizar por el software.
- Una característica de calidad no funcional, tales como la fiabilidad o la usabilidad.
- La estructura o arquitectura del software o sistema.
- Cambios asociados, es decir, confirmar que se han solucionado los defectos (pruebas de confirmación) y localizar cambios no intencionados (pruebas de regresión).

Los tipos de prueba a aplicar para la validación del módulo son funcionales y de regresión.

#### ➤ Pruebas funcionales

Las funciones que un sistema, subsistema o componente deben llevar a cabo pueden describirse en productos de trabajo tales como una especificación de requisitos, casos de uso o una especificación funcional, o incluso pueden no estar documentadas. Las funciones son “lo que” el sistema hace. Las pruebas funcionales se basan en funciones y prestaciones (descritas en documentos o entendidas por los probadores) y en su interoperabilidad con sistemas específicos, y pueden llevarse a cabo en todos los niveles de pruebas.

#### ➤ Pruebas de regresión

Una vez detectado y corregido un defecto, el software debe volver a probarse para confirmar que el defecto original ha sido corregido con éxito. A esto se le denomina confirmación. La depuración (localizar y corregir defectos) es una actividad de desarrollo, no una actividad de pruebas. Las pruebas de regresión son pruebas reiteradas de un programa ya probado, después de haber sido modificado, con vistas a localizar defectos surgidos o no descubiertos como resultado del cambio o de los cambios. Estos defectos pueden estar en el software objeto de las pruebas, o en cualquier otro componente de software asociado o no asociado. Se realizan cuando el software, o su entorno, sufren modificaciones. El alcance de las pruebas de regresión depende del riesgo de no encontrar defectos en el software que antes funcionaba.

### 3.5.4 Métodos de prueba

Los métodos de prueba se encargan de definir la estrategia a seguir en cuanto a la verificación y validación del módulo diseñado para descubrir fallos. Los métodos empleados fueron las pruebas de caja negra y caja blanca.

### 3.5.5 Caso de prueba

Un caso de prueba es una especificación de un caso para probar el sistema, incluyendo qué probar, con qué entradas y resultados y bajo qué condiciones. Su principal objetivo es obtener un conjunto de pruebas que tengan una mayor probabilidad de descubrir los defectos del software (Pressman, 2010).

### 3.6 Desarrollo de las pruebas internas

En esta etapa el desarrollador se realizan pruebas al módulo mediante los métodos: Pruebas de caja blanca, haciendo uso de la técnica: Prueba del camino básico y Prueba de caja negra utilizando la técnica de partición de equivalencia. A continuación, se explican dichas pruebas.

#### 3.6.1 Pruebas de caja negra

Las pruebas de caja negra, también llamadas pruebas de comportamiento, se enfocan en los requerimientos funcionales del software; es decir, las técnicas de prueba de caja negra le permiten derivar conjuntos de condiciones de entrada que revisarán por completo todos los requerimientos funcionales para un programa. Las pruebas de caja negra no son una alternativa para las técnicas de caja blanca. En vez de ello, es un enfoque complementario que es probable que descubra una clase de errores diferente que los métodos de caja blanca (Pressman, 2010).

- 1) Las pruebas de caja negra intentan encontrar errores en las categorías siguientes:
- 2) Funciones incorrectas o faltantes,
- 3) Errores de interfaz,
- 4) Errores en las estructuras de datos o en el acceso a bases de datos externas,
- 5) Errores de comportamiento o rendimiento
- 6) Errores de inicialización y terminación.

Se utiliza el método de caja negra aplicando la técnica partición de equivalente, la que 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 (Pressman, 2010).

#### 3.6.2 Pruebas de caja blanca

La prueba de caja blanca, en ocasiones llamada prueba de caja de vidrio, es una filosofía de diseño de casos de prueba que usa la estructura de control descrita como parte del diseño a nivel de componentes para derivar casos de prueba (Pressman, 2010).

Al hacer uso de la prueba de caja blanca, se puede derivar casos de prueba que:

- 1) Garanticen que todas las rutas independientes dentro de un módulo se revisaron al menos una vez,
- 2) Revisen todas las decisiones lógicas en sus lados verdadero y falso,
- 3) Ejecuten todos los bucles en sus fronteras y dentro de sus fronteras operativas y

4) Revisen estructuras de datos internas para garantizar su validez.

Se realiza la técnica de prueba de ruta o trayectoria básica que permite al diseñador de casos de prueba derivar una medida de complejidad lógica de un diseño de procedimiento y usar esta medida como guía para definir un conjunto básico de rutas de ejecución. Los casos de prueba derivados para revisar el conjunto básico tienen garantía para ejecutar todo enunciado en el programa, al menos una vez durante la prueba.

### **Paso de la técnica trayectoria básica**

1. Se escoge el diseño o el código de la aplicación como base, y se dibuja el correspondiente grafo de flujo. Se obtiene la complejidad ciclomática del grafo de flujo.
2. Se determina la complejidad ciclomática del grafo resultante, posteriormente se determinan los casos de pruebas que permitan la ejecución de los caminos anteriores.
3. Se establece un conjunto básico de caminos linealmente independientes.
4. Se elaboran los casos de pruebas que permitirán la ejecución de cada camino del conjunto básico.

### **Notación del grafo de flujo**

- Nodo del grafo de flujo(N): Es un círculo que representa las sentencias procedimentales.
- Aristas(A): Son las flechas o enlaces que representan el flujo de control del grafo.
- Regiones: Son las áreas delimitadas por las aristas y nodos. Al realizar el conteo de las regiones también se incluye el área exterior del grafo.

Complejidad ciclomática: es una métrica que proporciona una medición cuantitativa de la complejidad lógica de un programa. El valor calculado define el número de caminos independientes del conjunto básico de un programa. Esto indica el límite superior para el número de pruebas que se deben realizar, para asegurar que se ejecuta cada sentencia al menos una vez. La complejidad ciclomática tiene fundamentos en la teoría de gráficos y proporciona una medición de software extremadamente útil. La complejidad se calcula en una de tres formas:

- El número de regiones del gráfico de flujo corresponde a la complejidad ciclomática.
- La complejidad ciclomática  $V(G)$  para un gráfico de flujo  $G$  se define como  $V(G) = A - N + 2$  donde  $A$  es el número de aristas del gráfico de flujo y  $N$  el número de nodos del gráfico de flujo.
- La complejidad ciclomática  $V(G)$  para un gráfico de flujo  $G$  también se define como  $V(G) = P + 1$  donde  $P$  es el número de nodos predicado contenidos en el gráfico de flujo  $G$  (Pressman, 2010).

A continuación, se muestra un fragmento de código perteneciente al algoritmo desarrollado:

```

function getTablaDeLista() {
1  var listaNodos = new Array();
   var folderLista = companyhome.childByNamePath("Sitios/recomendaciones/documentLibrary/listaDeItems");
   var nodes = folderLista.children;
2  if (nodes) {
3    for (var i = 0; i < nodes.length; i++) {
4      if (nodes[i].properties["rec:uuidCreadoPor"] == person.id) {
5        listaNodos[listaNodos.length] = nodes[i];
       }
     }
6  return listaNodos;
}

```

Ilustración 12: Código de la función getTablaDeLista()

A continuación, se procede a la elaboración del grafo de flujo:

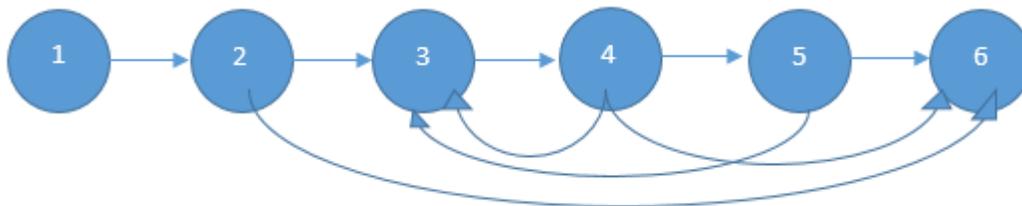


Ilustración 13: Gráfico de flujo de la función getTablaDeLista()

**Complejidad ciclomática**

$V(G) = A - N + 2$

$V(G) = 9 - 6 + 2$

$V(G) = 5$

$V(G) = P + 1$

$V(G) = 4 + 1$

$V(G) = 5$

$V(G) = \text{Regiones}$

$V(G) = 5$

**Caminos independientes**

Camino 1: 1,2,3,4,5,6

Camino 2: 1,2,6

Camino 3: 1,2,3,4,3,4,5,6

Camino 4: 1,2,3,4,6

Camino 5: 1,2,3,4,5,3,4,5,6

A continuación, se definen el caso de prueba para el camino básico 1 y se muestra el resultado de la prueba aplicada al camino.

Tabla 6: Casos de prueba de caja blanca para el camino básico 1

<b>Caso de prueba 1</b>	
<b>Descripción</b>	A partir de guardar en una lista de nodos, permite obtener una lista de nodos de tipo Lista de Ítems.
<b>Condición de ejecución</b>	La carpeta de nodos no puede estar vacía.
<b>Datos de entrada</b>	N/A
<b>Tipo de datos de esperado</b>	Lista de nodos de tipo Lista de Ítems
<b>Evaluación del caso de prueba</b>	Satisfactoria

**3.7 Resultados de las pruebas internas**

Una vez terminadas las pruebas internas ejecutadas por el desarrollador, se fue capaz comprobar que el módulo cumple con los requisitos funcionales definidos por la investigación. A continuación, se muestra la relación de no conformidades detectadas, que proceden y no proceden por iteración.

Tabla 7: Resultado de las pruebas internas ejecutadas por el desarrollador

<b>Iteración</b>	<b>NC detectadas</b>	<b>Asociadas a</b>	<b>NC proceden</b>	<b>NC no proceden</b>
I	10	Interfaz Ortografía Funcionalidad	10	0
II	4	Interfaz Funcionalidad	4	0
III	0		0	0

En la tabla anterior se puede apreciar que se realizaron 3 iteraciones de pruebas internas ejecutadas por el desarrollador. En la primera iteración se detectaron 10 no conformidades: 7 asociadas a errores de interfaz y de funcionalidad y 3 a errores ortográficos. Todas estas no conformidades fueron resueltas en dicha iteración. En la segunda iteración fueron detectadas 4 no conformidades. Estas no conformidades estaban asociadas a errores de interfaz y de funcionalidad, siendo satisfactoriamente solucionadas. En una tercera iteración no se detectaron no conformidades.

### **Conclusiones del capítulo**

- El diagrama de despliegue diseñado proporcionó un mejor entendimiento de cómo funciona la comunicación entre las distintas aplicaciones sentando las bases para la etapa de construcción del módulo.
- Se detalló la estrategia a seguir para desarrollarle las pruebas al Módulo Recomendaciones y comprobar el correcto funcionamiento del mismo.
- Las pruebas internas realizadas permitieron corregir errores detectados durante la fase de construcción del software, logrando la correcta implementación de los requisitos funcionales.

### **CONCLUSIONES GENERALES**

Durante el desarrollo de la investigación se planteó la necesidad de desarrollar un Módulo Recomendaciones de XABAL Repxos en la tecnología de Alfresco Community 5.2, dándole así cumplimiento al objetivo propuesto de la siguiente forma:

1. Se realizó un estudio de los referentes teóricos y metodológicos a tener en cuenta para abordar la solución del problema planteado relacionado con el Sistema de Recomendación en el sistema de XABAL Repxos con la tecnología de Alfresco.
2. Se evidenció el estado de la gestión documental de los usuarios del sistema XABAL Repxos.
3. Se implementó el Módulo Recomendaciones contribuyendo a la recuperación de los ítems más relevantes para los usuarios entre el sistema XABAL REPXOS y la plataforma de Alfresco, según sus preferencias.
4. 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, el autor realiza las siguientes recomendaciones:

- Incorporar en el módulo la posibilidad de que un usuario, una vez recibida la recomendación, pueda reenviársela a otros usuarios sin tener que crear una nueva.

### REFENRECIAS BIBLIOGRÁFICAS

1. ADOMAVICIUS, G. y TUZHILIN, A., 2005. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. . S.l.: s.n., pp. 734–749.
2. ALBÍN RODRÍGUEZ, A.P., 2009. *SISTEMA DE RECOMENDACIÓN COLABORATIVO BASADO EN ALGORITMOS DE FILTRADO MEJORADOS*. Proyecto Fin de carrera. Escuela Politécnica Superior de Jaén: Universidad de Jaén.
3. ALFRESCO COMMUNITY, 2013. Web Scripts | Alfresco Community. [en línea]. [Consulta: 20 mayo 2019]. Disponible en: <https://community.alfresco.com/docs/DOC-5166-web-scripts>.
4. Apache Tomcat® - Welcome! [en línea], 2015. [Consulta: 20 mayo 2019]. Disponible en: <http://tomcat.apache.org/>.
5. ARTEAGA, S., 2017. Así funciona el sistema de recomendaciones de Netflix. *ComputerHoy* [en línea]. [Consulta: 10 mayo 2019]. Disponible en: <https://computerhoy.com/noticias/apps/asi-funciona-sistema-recomendaciones-netflix-66807>.
6. AXURE, 2014. Herramientas de diseño » AXURE, Diseño de wireframes y Prototipos. [en línea]. [Consulta: 20 febrero 2019]. Disponible en: <http://multimedia.uoc.edu/blogs/labeines/es/prototipat/axure-disseny-de-wireframes-i-prototips/>.
7. BREESE, J., HECKERMAN, D. y KADIE, C., 1998. Empirical analysis of predictive algorithms for collaborative filtering. *Proc. of the 14th Annual Conf. on Uncertainty in Artificial Intelligence*. S.l.: s.n., pp. 43–52.
8. EMATIZ, 2013. Tipos de contenidos personalizados. . S.l.: ematiz.
9. FREEMAKER, 2014. What is Apache FreeMarker™? *Apache FreeMarker™* [en línea]. [Consulta: 20 mayo 2019]. Disponible en: <https://freemarker.apache.org/index.html>.
10. GASTAMINZA, F. del V., 2006. *Documentos digitales: hipertexto y edición digital*. 2006. S.l.: Universidad de Complutense Madrid.
11. GAUCHAT, J.D., 2012. *El gran libro de HTML5, CSS3 y Javascript*. Primera. Barcelona: MARCOMBO, S.A. ISBN 978-84-267-1782-.

## REFENRECIAS BIBLIOGRÁFICAS

12. HERLOCKER, J.L., KONSTAN, J.A. y RIEDL, J., 2002. An Empirical Analysis of Design Choices in Neighborhood-Based Collaborative Filtering Algorithms. *Information Retrieval*. S.l.: s.n., pp. 287-310.
13. HILL, W., STEAD, L., ROSENSTEIN, M. y FURNAS, G., 1995. Recommending and evaluating choices in a virtual community of use. *Proc. of the SIGCHI Conf. on Human Factors in Computing Systems*. New York: ACM Press/Addison-Wesley publishing Co.,
14. ISTQB, colectivo de autores, 2011. *International Software Testing Qualification Boards Probador Certificado Programa de estudio de nivel básico*. 2011. S.l.: s.n.
15. JARROBA, 2015. ¿Que son los Sistemas de Recomendación? [en línea]. [Consulta: 10 mayo 2019]. Disponible en: <https://jarroba.com/que-son-los-sistemas-de-recomendacion/>.
16. LARA, M., 2013. Pruebas de software. . S.l.: s.n., pp. 1-34.
17. LARMAN, C., 2003. *UML y Patrones. Introducción al análisis y diseño orientado a objetos*. 2da. Madrid: Pearson Educación.
18. LINDEN, G., SMITH, B. y YORK, J., 2003. *Recommendations. Item-to-Item Collaborative Filtering*. 2003. S.l.: s.n.
19. MAIDA, E.G. y PACIENZA, J., 2015. *Metodologías de desarrollo de software*. 2015. S.l.: Biblioteca digital de la universidad católica de Argentina.
20. MUÑOZ SERAFIN, M., 2018. *Introducción al desarrollo de aplicaciones N-Capas con tecnologías Microsoft*. 2018. S.l.: s.n.
21. O'REILLY MEDIA, I., 2014. A Technical Introduction to XML. [en línea]. [Consulta: 20 mayo 2019]. Disponible en: <https://www.xml.com/pub/a/98/10/guide0.html#AEN58>.
22. OBS, B., 2018. *¿Cuál es la utilidad de la matriz de trazabilidad?* 2018. S.l.: OBS Business School.
23. PANDORA, 2018. Cómo Pandora construyó un mejor motor de recomendaciones. *SearchDataCenter&nbsp;en&nbsp;Español* [en línea]. [Consulta: 21 mayo 2019]. Disponible en: <https://searchdatacenter.techtarget.com/es/cronica/Como-Pandora-construyo-un-mejor-motor-de-recomendaciones>.

## REFENRECIAS BIBLIOGRÁFICAS

24. PARADIGM, V., 2013. *Visual paradigm for uml. Visual Paradigm for UML-UML tool for software application development*. 2013. S.l.: s.n.
25. PRESSMAN, R.S., 2010. *Ingeniería del software.UN ENFOQUE PRÁCTICO*. 7. University of Connecticut: MCGRAW-HILL INTERAMERICANA EDITORES, S.A. ISBN 978-607-15-0314-5.
26. RESNICK, P. y VARIAN, H.R., 1997. Recommender Systems. *Communications of the ACM*. S.l.: s.n., pp. 1–3.
27. RICCI, F., ROKACH, L., SHAPIRA, B. y KANTOR, P.B., 2011. *Recommender Systems Handbook*. New York: Springer. ISBN 978-0-387-85819-7.
28. RODRÍGUEZ SÁNCHEZ, T., 2015. *Metodología de desarrollo para la Actividad productiva de la UCI*. 2015. S.l.: s.n.
29. RUIZ, F., 2017. *Arquitectura Física del Sistema (en desarrollo OO)*. 2017. S.l.: s.n.
30. RUIZ RICARDO, B. y VAILLANT VALDÉZ, C. de la C., 2015. *Módulo Recomendaciones para el sistema Repxos*. S.l.: Universidad de las Ciencias Informáticas.
31. SANTIAGO, 2009. *GUÍA PARA LA CREACIÓN DE METADATOS USANDO DUBLIN CORE*. 2009. S.l.: Comité de Metadatos de la Biblioteca Nacional de Chile.
32. Sistemas de recomendación en eCommerce: claves para potenciar tus ventas. *Marketing 4 Ecommerce - Tu revista de marketing online para e-commerce* [en línea], 2017. [Consulta: 10 mayo 2019]. Disponible en: <https://marketing4ecommerce.net/sistemas-de-recomendacion-en-ecommerce/>.
33. SOMMERVILLE, I., 2011. *Ingeniería de Software*. México: Pearson Education. ISBN 978-607-32-0603-7.
34. STRATEBI, 2014. Alfresco Gestión Documental. [en línea]. [Consulta: 20 mayo 2019]. Disponible en: <http://www.stratebi.com/alfresco>.
35. TEXIER, J., DE GIUSTI, M. y OVIEDO, N., 2010. *El Uso de Repositorios y su Importancia para la Educación en Ingeniería*. 2010. S.l.: s.n.

## *REFENRECIAS BIBLIOGRÁFICAS*

36. THE POSTGRESQL GLOBAL DEVELOPMENT GROUP, 2012. *PostgreSQL 9.2.1 Documentation*. WhitePaper. California: University of California: s.n.
37. VALENCIA, L., 2017. VS code: 1 introduccion. *Luis Valencia - Sharepoint/Office365 MVP* [en línea]. [Consulta: 14 mayo 2019]. Disponible en: <http://www.luisevalencia.com/2017/05/01/vscode1introduccion/>.
38. ZAMORA CESÉ, Y., 2012. *Sistema de recomendación para el catálogo en línea de la biblioteca de la Universidad de las Ciencias Informáticas*. Trabajo de diploma para optar por el título de Ingeniero en Ciencias Informáticas. La Habana: Universidad de las Ciencias Informáticas.

**BIBLIOGRAFÍA**

1. Algunos conceptos básicos en el modelado de contenidos de Alfresco - zylk. [en línea], [sin fecha]. [Consulta: 14 febrero 2019]. Disponible en: <https://www.zylk.net/en/web-2-0/blog/-/blogs/algunos-conceptos-basicos-en-el-modelado-de-contenidos-de-alfresco>.
2. ALBÍN RODRÍGUEZ, A.P., 2009. *SISTEMA DE RECOMENDACIÓN COLABORATIVO BASADO EN ALGORITMOS DE FILTRADO MEJORADOS*. Proyecto Fin de carrera. Escuela Politécnica Superior de Jaén: Universidad de Jaén.
3. ALFRESCO, 2013. La plataforma empresarial digital de Alfresco | Alfresco. [en línea]. [Consulta: 31 enero 2019]. Disponible en: <https://www.alfresco.com/es/plataforma#>.
4. ALFRESCO COMMUNITY, 2013. Web Scripts | Alfresco Community. [en línea]. [Consulta: 20 mayo 2019]. Disponible en: <https://community.alfresco.com/docs/DOC-5166-web-scripts>.
5. BREESE, J., HECKERMAN, D. y KADIE, C., 1998. Empirical analysis of predictive algorithms for collaborative filtering. *Proc. of the 14th Annual Conf. on Uncertainty in Artificial Intelligence*. S.l.: s.n., pp. 43–52.
6. Características de Alfresco - CyberPymes. [en línea], [sin fecha]. [Consulta: 20 mayo 2019]. Disponible en: <http://www.cyberpymes.com/aplicaciones-a-medida/alfresco-caracteristicas>.
7. JARROBA, 2015. ¿Que son los Sistemas de Recomendación? [en línea]. [Consulta: 10 mayo 2019]. Disponible en: <https://jarroba.com/que-son-los-sistemas-de-recomendacion/>.
8. LINDEN, G., SMITH, B. y YORK, J., 2003. *Recommendations. Item-to-Item Collaborative Filtering*. 2003. S.l.: s.n.
9. MARTÍNEZ MIMBRERA, F.J., 2008. *Sistema de Recomendación Actualizable y con Gestión de usuarios* [en línea]. 2008. S.l.: s.n. Disponible en: [http://sinbad2.ujaen.es/cod/archivosPublicos/pfc/pfc\\_francisco\\_martinez.pdf](http://sinbad2.ujaen.es/cod/archivosPublicos/pfc/pfc_francisco_martinez.pdf).
10. Outbound SMTP configuration properties | Alfresco Documentation. [en línea], [sin fecha]. [Consulta: 20 mayo 2019]. Disponible en: <http://docs.alfresco.com/community/concepts/email-outboundsmtp-props.html>.
11. RESNICK, P. y VARIAN, H.R., 1997. Recommender Systems. *Communications of the ACM*. S.l.: s.n.,

pp. 1–3.

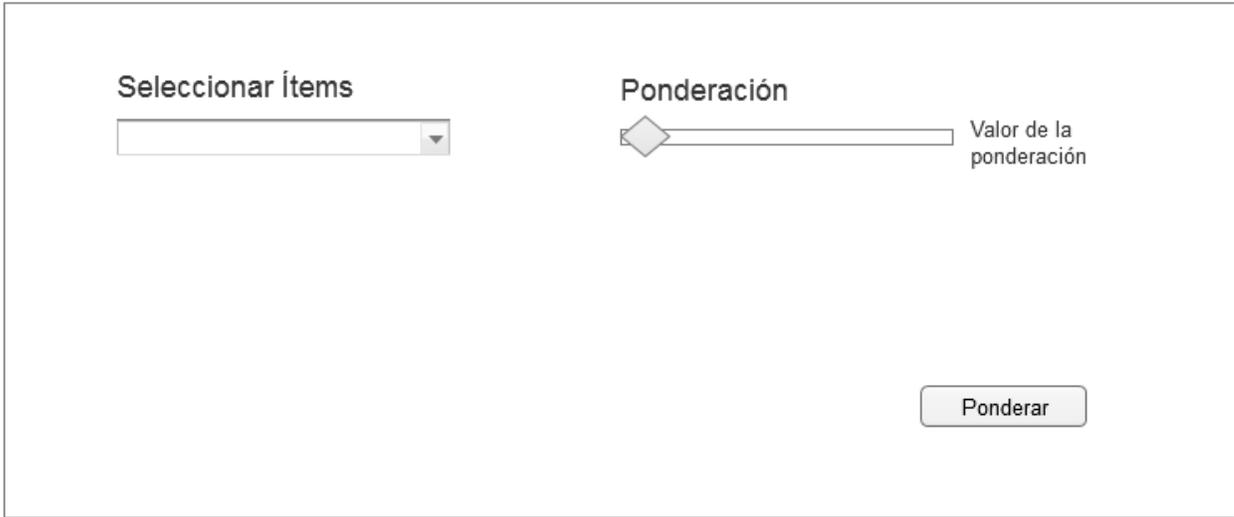
12. RICCI, F., ROKACH, L., SHAPIRA, B. y KANTOR, P.B., 2011. *Recommender Systems Handbook*. New York: Springer. ISBN 978-0-387-85819-7.
13. RUIZ RICARDO, B. y VAILLANT VALDÉZ, C. de la C., 2015. *Módulo Recomendaciones para el sistema XABAL Repxos*. S.I.: Universidad de las Ciencias Informáticas.
14. STRATEBI, 2014. Alfresco Gestión Documental. [en línea]. [Consulta: 20 mayo 2019]. Disponible en: <http://www.stratebi.com/alfresco>.
15. PRESSMAN, R.S., 2010. *Ingeniería del software. UN ENFOQUE PRÁCTICO*. 7. University of Connecticut: McGraw-Hill INTERAMERICANA EDITORES, S.A. ISBN 978-607-15-0314-5.
16. SOMMERVILLE, I., 2011. *Ingeniería de Software*. México: Pearson Education. ISBN 978-607-32-0603-7.
17. HERNANDEZ SAMPIERI, R., 2014. *Metodología de la Investigación*. 6. S.I.: MCGRAW-HILL / INTERAMERICANA EDITORES, S.A. DE C.V. ISBN 978-1-4562-2396-0.

## Anexo 1: Descripción de los CUS

### ✓ CUS 1: Ponderar ítem

Tabla 8: Descripción del CUS Ponderar Ítems

<b>Objetivo</b>	Ponderar uno o varios ítems, según la apreciación del usuario, en una escala del 1 al 5, donde 5 es la máxima calificación.	
<b>Actores</b>	Usuario	
<b>Resumen</b>	El caso de uso se inicia cuando el usuarios desea darle una clasificación a uno o varios ítems según su apreciación.	
<b>Complejidad</b>	Media	
<b>Prioridad</b>	Alta	
<b>Precondiciones</b>	No aplica	
<b>Postcondiciones</b>	Uno o varios ítems fueron ponderados.	
<b>Flujo de eventos</b>		
<b>Flujo básico Ponderar Ítems</b>		
	<b>Actor</b>	<b>Sistema</b>
1.	Selecciona la opción "Ponderar Ítems"	
2.		Muestra un formulario solicitando los siguientes datos: <ul style="list-style-type: none"> <li>- Ítems a ponderar</li> <li>- Ponderación</li> </ul> Muestra la opción: <ul style="list-style-type: none"> <li>- Ponderar</li> </ul>
3.	Introduce los datos de la ponderación.	
4.	Da clic en la opción "Ponderar"	
5.		Comprueba que se hayan introducidos los datos obligatorios.
6.		Muestra un mensaje: "Se ponderò correctamente".
7.		Termina el caso de uso.
<b>Flujos alternos</b>		
<b>5a. Los datos son incorrectos.</b>		
	<b>Actor</b>	<b>Sistema</b>
1.		Comprueba que se han introducido datos incorrectos.
2.		Señala los campos que deben ser llenados, indicando que existen errores en el formulario.
<b>Relaciones</b>	<b>CU incluidos</b>	N/A
	<b>CU extendidos</b>	N/A

Requisitos no funcionales	N/A
Asuntos pendientes	N/A
<b>Prototipo elemental de interfaz gráfica de usuario</b>	
	

✓ **CUS 3: Recomendar ítems de forma automática.**

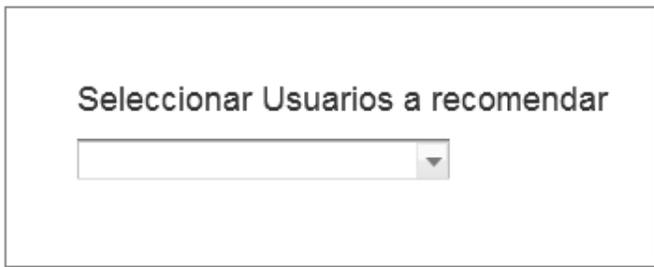
Tabla 9: Descripción del CUS Recomendar ítems de forma automática

<b>Objetivo</b>	Recomendar ítems de forma automática	
<b>Actores</b>	Usuario	
<b>Resumen</b>	El caso de uso inicia cuando el usuario pondera ítems y a través de esto se realiza la recomendación automática a otros usuarios.	
<b>Complejidad</b>	Alta	
<b>Prioridad</b>	Alta	
<b>Precondiciones</b>	Los ítems han sido ponderados.	
<b>Postcondiciones</b>	Uno o varios ítems han sido recomendados.	
<b>Flujo de eventos</b>		
<b>Flujo básico Recomendar ítems de forma automática</b>		
	<b>Actor</b>	<b>Sistema</b>
1.	Pondera ítems	
2.		Recomienda ítems
3.		Termina el caso de uso
<b>Relaciones</b>	<b>CU incluidos</b>	Pondera ítems. <u>Ver CU Ponderar ítems</u>
	<b>CU extendidos</b>	N/A
<b>Requisitos no funcionales</b>	N/A	

<b>Asuntos pendientes</b>	N/A
---------------------------	-----

✓ **CUS 4: Buscar usuario a recomendar.**

Tabla 10: Descripción del CUS Buscar usuario a recomendar

<b>Objetivo</b>	Hacer una búsqueda de los usuarios a recomendar.	
<b>Actores</b>	Usuario	
<b>Resumen</b>	El caso de uso inicia cuando el usuario, para recomendar ítems, realiza una búsqueda de los usuarios del sistema.	
<b>Complejidad</b>	Media	
<b>Prioridad</b>	Alta	
<b>Precondiciones</b>	No aplica	
<b>Postcondiciones</b>	Uno o varios usuarios fueron seleccionados a recomendar.	
<b>Flujo de eventos</b>		
<b>Flujo básico Buscar usuario a recomendar</b>		
	<b>Actor</b>	<b>Sistema</b>
1.	Busca el o los usuarios a recomendar.	
2.		A través de un campo de selección múltiple muestra los usuarios autenticados.
3.	Selecciona el o los usuarios a recomendar.	
4.		Termina el caso de uso.
<b>Prototipo elemental de interfaz gráfica de usuario</b>		
		

✓ **CUS 5: Gestionar recomendación de ítem a usuario.**

Tabla 11: Descripción del CUS Gestionar recomendación de ítem a usuario

<b>Objetivo</b>	Gestionar las recomendaciones que se le han hecho al usuario ya se de forma manual o automática.
<b>Actores</b>	Usuario
<b>Resumen</b>	El caso de uso inicia cuando el usuario necesita revisar las recomendaciones que le han hecho.
<b>Complejidad</b>	Alta

<b>Prioridad</b>	Alta	
<b>Precondiciones</b>	Se hizo una recomendación.	
<b>Postcondiciones</b>	Muestra las recomendaciones que le han hecho.	
<b>Flujo de eventos</b>		
<b>Flujo básico Gestionar recomendación de ítem a usuario.</b>		
	<b>Actor</b>	<b>Sistema</b>
1.	Selecciona la opción "Notificaciones"	
2.		- Muestra una tabla con las recomendaciones de ítems.
3.		Permite realizar las siguientes acciones: <ul style="list-style-type: none"> <li>- Mostrar ítems. Ver Sección 1: Mostrar ítems de la recomendación a usuario.</li> <li>- Eliminar recomendación de ítem. Ver Sección 2: Eliminar recomendación de ítem a usuario.</li> </ul>
4.		Termina el caso de uso.
<b>Sección 1: "Mostrar ítems de la recomendación a usuario"</b>		
<b>Flujo básico Mostrar ítems de la recomendación a usuario</b>		
	<b>Actor</b>	<b>Sistema</b>
1.	Selecciona la acción "Mostrar ítems"	
2.		Abre un modal mostrando los ítems de la recomendación. Muestra la siguiente opción: <ul style="list-style-type: none"> <li>- Cerrar</li> </ul>
1.	Selecciona la opción "Cerrar".	
2.		Cierra el modal y retorna a la interfaz que le dio origen.
<b>Sección 2: "Eliminar recomendación de ítem a usuario"</b>		
<b>Flujo básico Eliminar recomendación de ítem a usuario</b>		
	<b>Actor</b>	<b>Sistema</b>
1.	Selecciona la acción eliminar recomendación.	
2.		Muestra un mensaje de confirmación con las siguientes opciones: <ul style="list-style-type: none"> <li>- Aceptar</li> <li>- Cancelar</li> </ul>
3.	Selecciona la opción "Aceptar".	
4.		Cierra el mensaje de confirmación.
<b>Flujos alternos</b>		
<b>2ª Opción "Cancelar"</b>		

	Actor	Sistema									
1.	Selecciona la opción "Cancelar"										
2.		Cierra el mensaje de confirmación.									
<b>Relaciones</b>	<b>CU incluidos</b>	N/A									
	<b>CU extendidos</b>	N/A									
<b>Prototipo elemental de interfaz gráfica de usuario</b>											
<table border="1" style="margin: auto;"> <thead> <tr> <th>Recomendado por:</th> <th>Descripción</th> <th>Acciones</th> </tr> </thead> <tbody> <tr> <td> </td> <td> </td> <td> </td> </tr> <tr> <td> </td> <td> </td> <td> </td> </tr> </tbody> </table>			Recomendado por:	Descripción	Acciones						
Recomendado por:	Descripción	Acciones									

✓ **CUS 6: Recibir notificación recomendación por correo**

Tabla 12: Descripción del CUS Recibir notificación por correo

<b>Objetivo</b>	Notificar una recomendación por correo a los usuarios seleccionados.	
<b>Actores</b>	Usuario	
<b>Resumen</b>	El caso de uso inicia cuando el usuarios realiza una recomendación, automáticamente se le envía un correo notificándole a los usuarios que tienen una recomendación nueva.	
<b>Complejidad</b>	Alta	
<b>Prioridad</b>	Alta	
<b>Precondiciones</b>	Se ha realizado una recomendación.	
<b>Postcondiciones</b>	Se envió el correo de notificación.	
<b>Flujo de eventos</b>		
<b>Flujo básico &lt;Nombre del flujo básico&gt;</b>		
	<b>Actor</b>	<b>Sistema</b>
1.		Envía un correo de notificación.
2.		Termina el caso de uso.
<b>Relaciones</b>	<b>CU incluidos</b>	Recomendar ítems de forma manual. <u>Ver CU Recomendar ítems de forma manual.</u> Recomendar ítems de forma automática. <u>Ver CU Recomendar ítems de forma automática.</u>
	<b>CU extendidos</b>	N/A
<b>Requisitos no funcionales</b>	N/A	
<b>Asuntos pendientes</b>	N/A	

## ✓ CUS 7: Gestionar lista de ítems

Tabla 13: Descripción del CUS Gestionar Lista de ítems

<b>Objetivo</b>	Crear, editar, eliminar y mostrar listas de ítems.	
<b>Actores</b>	Usuario	
<b>Resumen</b>	El caso de uso inicia cuando el usuario necesita gestionar una lista de ítems para acceder de manera más fácil a los ítems del sistema guardándolos en dicha lista, pudiendo editarla o eliminarla.	
<b>Complejidad</b>	Alta	
<b>Prioridad</b>	Alta	
<b>Precondiciones</b>	No aplica.	
<b>Postcondiciones</b>	La lista de ítems ha sido creada.	
<b>Flujo de eventos</b>		
<b>Flujo básico Gestionar lista de ítems</b>		
	<b>Actor</b>	<b>Sistema</b>
1.	Selecciona la opción gestionar lista de ítems.	
2.		<p>Muestra un formulario con los siguientes campos obligatorios:</p> <ul style="list-style-type: none"> <li>- Nombre de la lista</li> <li>- Tipo de visibilidad</li> <li>- Seleccionar ítem</li> </ul> <p>Muestra las siguientes opciones:</p> <ul style="list-style-type: none"> <li>- Aceptar</li> <li>- Cancelar</li> </ul> <p>Muestra una tabla con las listas creadas que permite realizar las siguientes acciones:</p> <ul style="list-style-type: none"> <li>- Editar lista. Ver sección 1: Editar lista.</li> <li>- Mostrar ítems. Ver sección 3: Mostrar ítems de la lista de ítems.</li> <li>- Eliminar lista. Ver sección 3: Eliminar lista.</li> </ul>
3.	Introduce los datos de los campos obligatorios.	
4.	Da clic en la opción "Aceptar".	
5.		Comprueba que se han llenado los campos correctamente.
6.		Muestra un mensaje: "Se ha creado una nueva lista".
7.		Añade la lista a la tabla de lista de ítems.
8.		Termina el caso de uso.

<b>Flujos alternos</b>		
<b>2a. Opción "Cancelar"</b>		
	<b>Actor</b>	<b>Sistema</b>
1.	Da clic en la opción "Cancelar".	
2.		Limpia los campos del formulario.
<b>Flujos alternos</b>		
<b>5a. Los datos son incorrectos</b>		
	<b>Actor</b>	<b>Sistema</b>
1.		Comprueba que se hayan introducidos datos incorrectos.
2.		Señala los campos que deben ser llenados indicando que existen errores en el formulario.
<b>Sección 1: "Editar lista"</b>		
<b>Flujo básico Editar lista</b>		
	<b>Actor</b>	<b>Sistema</b>
1.	Selecciona la acción "Editar lista".	
2.		Muestra los campos del formulario con los datos de la lista. Muestra las siguientes opciones: <ul style="list-style-type: none"> <li>- Aceptar</li> <li>- Cancelar</li> </ul>
3.	Modifica los campos "Nombre de la lista" y "Tipo de visibilidad" .	
4.	Añade o elimina ítems a la lista de ítems.	
5.	Da clic en la opción "Aceptar"	
6.		Comprueba que se han llenado los campos correctamente.
7.		Muestra un mensaje: "Se ha modificado la lista satisfactoriamente".
8.		Cambia la lista de la tabla de lista de ítems por la modificada.
<b>Flujos alternos</b>		
<b>5a. Los datos son incorrectos</b>		
	<b>Actor</b>	<b>Sistema</b>
1.		Comprueba que se hayan introducidos datos incorrectos.

2.		Señala los campos que deben ser llenados indicando que existen errores en el formulario.
<b>Flujos alternos</b>		
<b>2a. Opción “Cancelar”</b>		
	<b>Actor</b>	<b>Sistema</b>
1.	Da clic en la opción “Cancelar”.	
2.		Limpia los campos del formulario.
<b>Sección 2: “Mostrar ítems”</b>		
<b>Flujo básico Mostrar ítems de la lista de ítems</b>		
	<b>Actor</b>	<b>Sistema</b>
1.	Selecciona la acción “Mostrar ítems”.	
2.		Abre un modal mostrando los ítems de la recomendación. Muestra la siguiente opción: - Cerrar
3.	Selecciona la opción “Cerrar”.	
		Cierra el modal y retorna a la interfaz que le dio origen.
<b>Sección 3: “Eliminar lista”</b>		
<b>Flujo básico Eliminar lista</b>		
	<b>Actor</b>	<b>Sistema</b>
1.	Selecciona la acción “Eliminar lista”.	
2.		Muestra un mensaje de confirmación con las siguientes opciones: - Aceptar - Cancelar
3.	Selecciona la opción “Aceptar”	
		Elimina la lista de ítems.
<b>Flujos alternos</b>		
<b>2ª. Opción “Cancelar”</b>		
	<b>Actor</b>	<b>Sistema</b>
1.	Selecciona la opción “Cancelar”	
2.		Cierra el mensaje de confirmación.
<b>Relaciones</b>	<b>CU incluidos</b>	N/A
	<b>CU extendidos</b>	N/A
<b>Prototipo elemental de interfaz gráfica de usuario</b>		

**Nombre de la lista**

**Visibilidad**

Público  Privado

**Seleccionar ítems**

Nombre de la lista	Visibilidad	Acciones

## Anexo 2: Interfaces gráficas del Módulo Recomendaciones

Repositorio Digital

Recomendaciones

Inicio Recomendaciones Herramientas de adm... Sitios Mis ficheros
hiram Hernandez

Recomendaciones

- Ponderar Items
- Gestionar Lista de Items
- Recomendar
- Notificaciones 2

**Crear lista de ítems**

**Nombre de la lista**

**Tipo**

Público  Privado

**Seleccionar ítem**

Mostrar  elementos

Nombre de la lista	Tipo	Acciones
Configuración	Público	✎ 🗑️ 📄
lista 1	Privado	✎ 🗑️ 📄

Mostrando 1 página(s) de 1
Anterior 1 Siguiente

Ilustración 14: Interfaz de Gestionar Lista de ítems

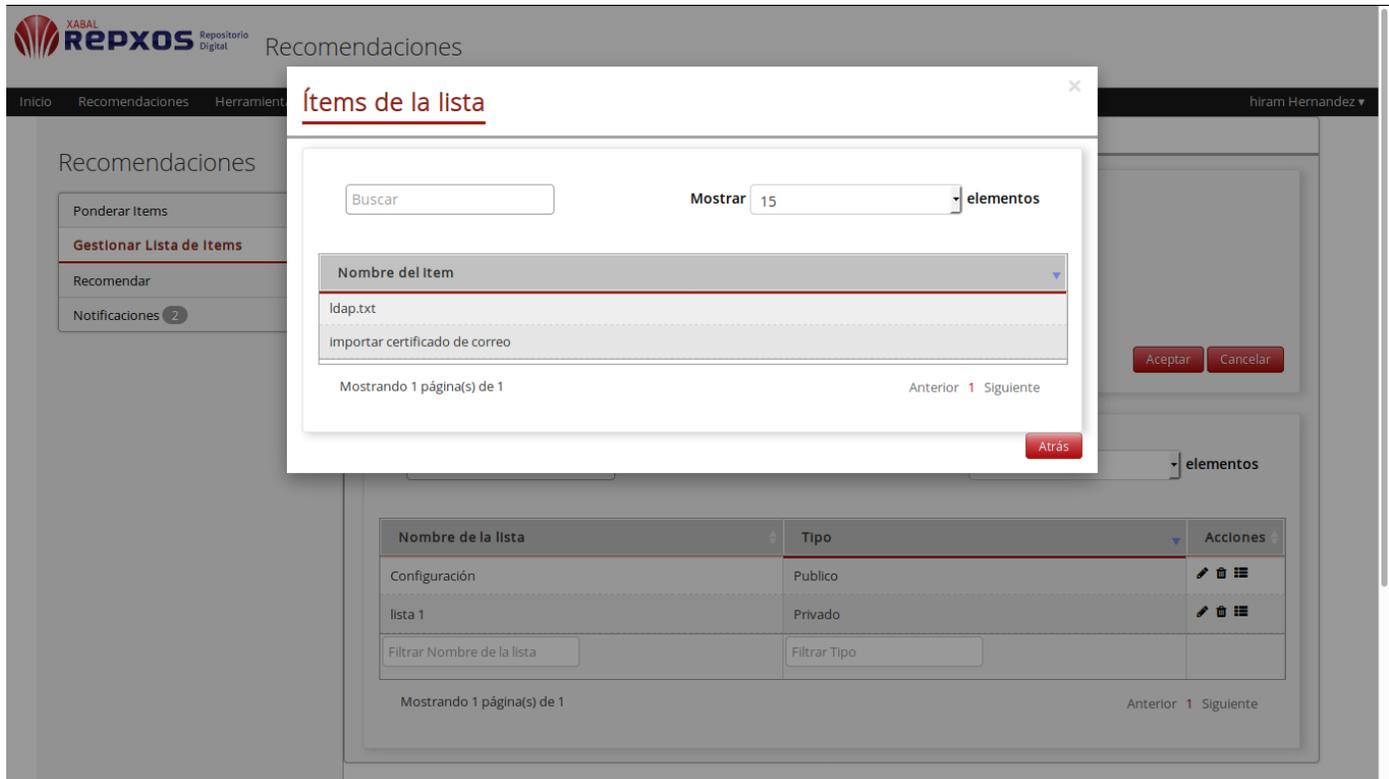


Ilustración 15: Interfaz de mostrar ítems de la lista de ítems

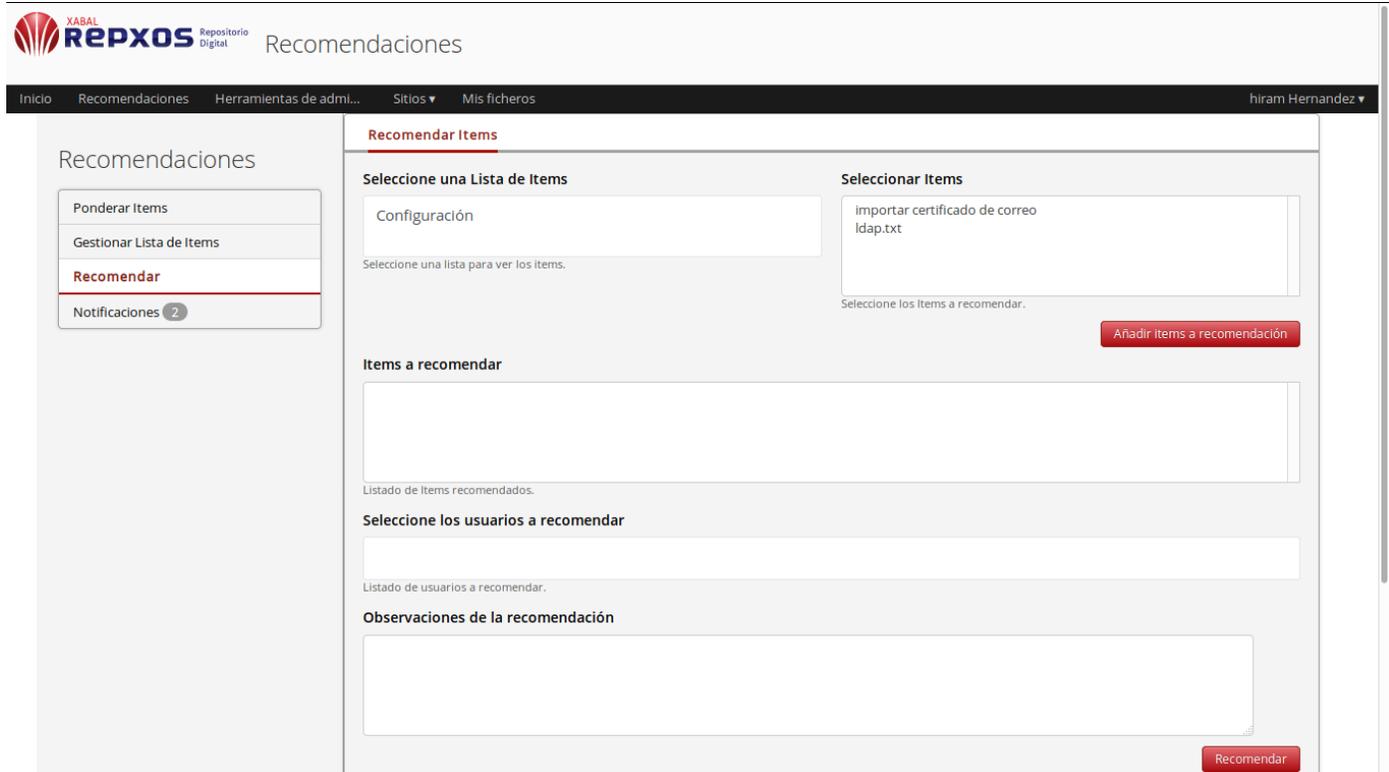


Ilustración 16: Interfaz de Recomendar ítems

Notificaciones

Buscar Mostrar 15 elementos

Recomendado por:	Descripcion	Acciones
Sistema	Recomendación nueva	
admin	Documentos interesantes	

Filtrar Recomendado por:  Filtrar Descripción

Mostrando 1 página(s) de 1 Anterior 1 Siguiete

Ilustración 17: Interfaz de mostrar recomendaciones hechas al usuario

Ítems de la recomendación

Buscar Mostrar 15 elementos

Nombre del Item	Acciones
ldap.txt	
Informe técnico PID VI.doc	

Mostrando 1 página(s) de 1 Anterior 1 Siguiete

[Atrás](#)

Ilustración 18: Mostrar ítems de la recomendación hecha al usuario



Ilustración 19: Interfaz de buscar usuario a recomendar

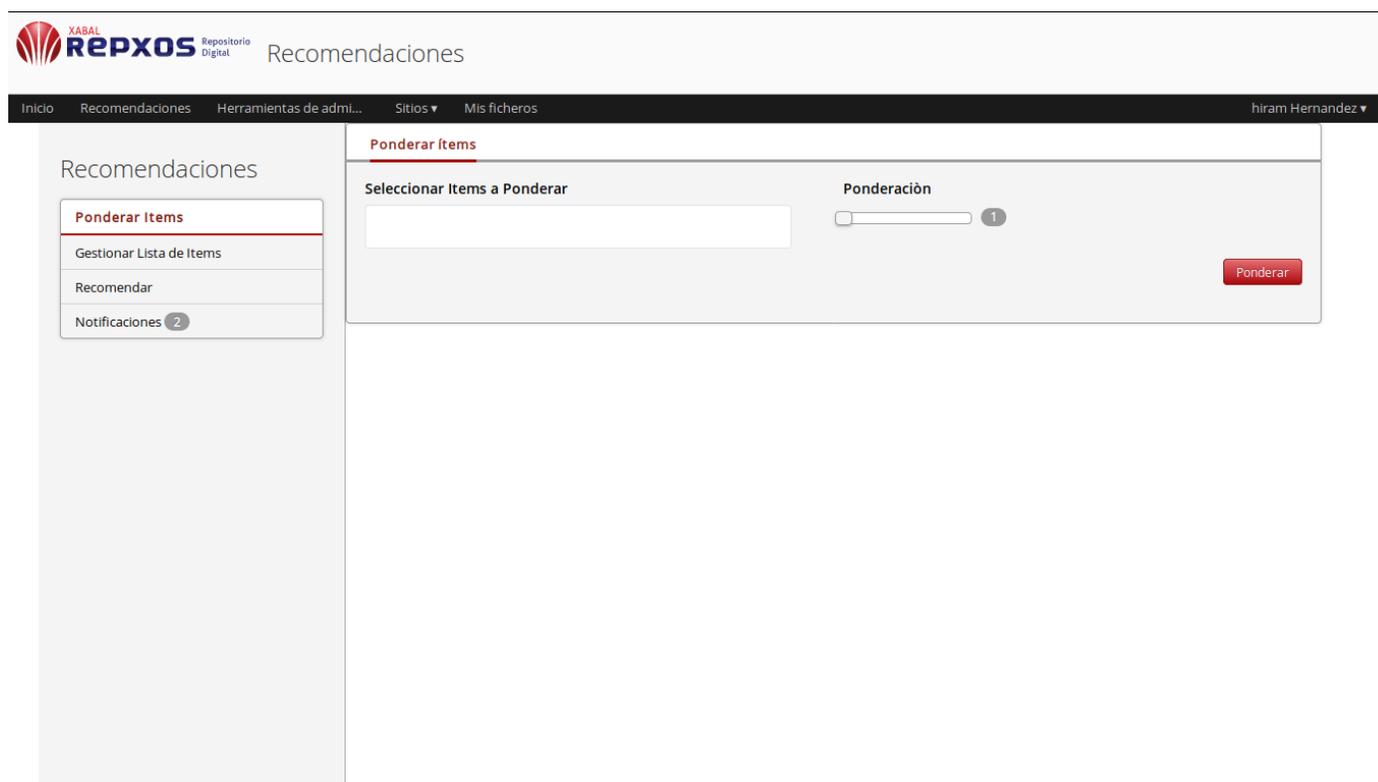


Ilustración 20: Interfaz de Ponderar ítems

## Anexo 3: Entrevista

La entrevista se realizó de manera informal. Las preguntas fluyeron en dependencia del conocimiento de los especialistas del centro sobre el modulo implementado.

1. ¿Cómo funciona el proceso de recomendaciones en el proyecto de Repositorio Institucional?
2. ¿Qué funcionalidades tiene el módulo recomendaciones del sistema XABAL Repxos?
3. ¿Cómo funciona el sistema de recomendación de filtrado colaborativo que utiliza el modulo recomendaciones del sistema XABA Repxos?

4. ¿Cuál es el objetivo de implementar este módulo?
5. ¿Qué datos se manejan en el módulo?
6. ¿Qué requisitos debe tener el modulo a desarrollar para que sea compatible con otros productos del centro?

Entrevista realizada a:

Analista del proyecto Repositorio Institucional: Gleidis Yuriannis Rosabal Espinosa

Desarrollador en el proyecto Repositorio Institucional: José Javier Hernández Benítez