



ESCUELA SUPERIOR POLITECNICA DEL LITORAL

**FACULTAD DE INGENIERIA EN ELECTRICIDAD Y
COMPUTACION**

TESIS DE GRADO

**“ANÁLISIS, DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA ADAPTIVO DE
RECOMENDACIÓN DE INFORMACIÓN BASADO EN MASHUPS”**

Previa a la obtención del Título de:

**INGENIERO EN COMPUTACIÓN
ESPECIALIZACIÓN SISTEMAS MULTIMEDIA**

Presentada por:

KATTY PATRICIA MIZHUERO CAÑAR

JORGE DANIEL BARRERA HEREDIA

GUAYAQUIL – ECUADOR

2009

AGRADECIMIENTO

A Dios y a mis padres por darme la oportunidad de realizar una carrera universitaria.

A mi compañera de tesis por compartir este trabajo y esta meta que alcanzamos juntos.

Jorge Barrera Heredia

*A Dios, por guiar mis pasos
constantemente y darme tantas
bendiciones.*

*A mis padres, por todo su amor,
sacrificio y palabras de aliento en pro
de alcanzar la meta de obtener una
carrera universitaria.*

*A mis amigos, por el apoyo
incondicional y la presión constante
para que culmine este trabajo de
tesis.*

Katty Mizhquero Cañar

DEDICATORIA

*A todas las personas que me
brindaron su apoyo a lo largo de mis
estudios y durante el desarrollo de la
tesis.*

Jorge Barrera Heredia

*Con cariño, a mis padres, quienes
confiaron en mí y me dieron su apoyo
incondicional para la feliz
culminación del presente trabajo de
tesis.*

Katty Mizhquero Cañar

TRIBUNAL DE GRADO

PRESIDENTE

Ing. Jorge Aragundi

DIRECTOR DE TESIS

PhD. Xavier Ochoa Chehab

VOCALES

PhD. Enrique Peláez

MSc. Cristina Abad

DECLARACIÓN EXPRESA

“La responsabilidad por el contenido de esta Tesis de Grado nos corresponde exclusivamente; y el patrimonio intelectual de la misma, a la Escuela Superior Politécnica del Litoral”

Reglamento de exámenes y títulos profesionales de la ESPOL

Katty Patricia Mizhquero Cañar

Jorge Daniel Barrera Heredia

RESUMEN

Actualmente, el crecimiento de la Web ha ocasionado que las personas tengan a su disposición mas información de la que pueden analizar. Esto causa sobrecarga de información, la cual puede ser evitada con soluciones alternativas como el uso de un sistema de recomendación. El objetivo de este trabajo es explorar el campo de los sistemas de recomendación y en él se detallan el análisis, diseño e implementación de un sistema adaptivo de recomendación de información basado en mashups.

En el Capítulo 1 se analiza el problema, revisado desde un punto de vista formal. Se incluye el marco teórico necesario para la comprensión del tema seleccionado, además; se dan antecedentes y las justificaciones para la realización de este proyecto de Tesis. De igual manera se definen también el alcance y los objetivos de este trabajo de investigación. En la parte final de este capítulo, se da la definición de los sistemas de recomendación, sus usos y características y se habla de los trabajos relacionados.

En el Capítulo 2 se detallan las herramientas disponibles para la solución del problema. Además se analizan las métricas empleadas en los

sistemas de recomendación y se realiza un análisis técnico de las alternativas existentes para cada uno de los componentes.

En el Capítulo 3 se establecen las necesidades y requerimientos de la solución. Luego se realiza el análisis de los casos de uso que incluye un diagrama general y una descripción de cada uno de ellos. Además analizamos 3 tipos de recomendación: recomendación social, basada en historial y por similitud de objetos. Finalmente hacemos la selección de las herramientas disponibles para implementar la solución al problema planteado, según las alternativas presentadas en el Capítulo 2.

En el Capítulo 4 se procede a realizar el diseño del sistema de recomendación con el que se propone cumplir con los objetivos propuestos al inicio de este trabajo de tesis. Se detalla la arquitectura del sistema, así como también el diseño de la base de datos y de las operaciones permitidas por el sistema.

En el Capítulo 5 se describe el proceso realizado en la implementación del Sistema, se mencionan los estándares de programación utilizados así como también los requerimientos de hardware y software necesarios para que el sistema sea puesto en producción. Se relata también los problemas presentados durante esta etapa.

En el Capítulo 6 se detallan las pruebas realizadas con el fin de medir el desempeño del sistema y la satisfacción del usuario.

Finalmente se llega a conclusiones, una vez terminado el trabajo propuesto y se dan las posibles recomendaciones. Se incluyen los apéndices que contendrán: el procedimiento para pruebas del sistema, el detalle del diagrama de clases del sistema.

ABREVIATURAS

API	Application programming interface
AUC	Area under curve
CD	Compact disc
CSS	Cascading style sheets
DOM	Document object model
H	High
HRI	Human recommender interaction
HTML	Hypertext markup language
HTTP	Hypertext transfer protocol
IDE	Integrated development environment
IDL	Interface description language
IF	Information filtering
IPC	Interprocess communication
IR	Information retrieval
L	Low
MAE	Mean absolute error
PC	Personal computer
RDF	Resource description framework
RMSE	Root mean squared error
ROC	Receiver operating characteristic
RPC	Remote procedure call
RSS	Really simple syndication
SGV	Scalable vector graphics
SOAP	Simple object access protocol
SWS	Semantic Web services
UML	Unified modeling language
URL	Uniform resource locator
W3C	World Wide Web Consortium
WSDL	Web services description language
XHTML	Extensible hypertext markup language
XML	Extensible markup language
XPCOM	Cross platform component object model
XPI	Cross-platform install
XUL	XML-based User-interface language
ZIP	Formato de compresión de archivos

SIMBOLOGÍA

\in = Pertenece a

\mathfrak{R} = Conjunto de números reales

E = Error

\overline{E} = Media del error

$\sum_{i=1}^n$ = Sumatoria desde 1 hasta n

$\text{Cos}(P, N)$ = Función de similitud del coseno, donde P es el vector de las calificaciones y N es el vector de las predicciones.

ÍNDICE GENERAL

“ANÁLISIS, DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA ADAPTIVO DE RECOMENDACIÓN DE INFORMACIÓN BASADO EN MASHUPS”	1
AGRADECIMIENTO.....	2
DEDICATORIA.....	4
TRIBUNAL DE GRADO	6
DECLARACIÓN EXPRESA	7
RESUMEN.....	8
ABREVIATURAS	11
SIMBOLOGÍA.....	12
ÍNDICE GENERAL	13
ÍNDICE DE FIGURAS	20
ÍNDICE DE TABLAS	22
INTRODUCCIÓN.....	23

CAPÍTULO 1	26
MARCO TEÓRICO	26
1.1 Definición del problema.....	27
1.2 Antecedentes.....	28
1.3 Justificación	29
1.4 Alcance y objetivos	30
1.4.1 Alcance	30
1.4.2 Objetivos.....	31
1.5 Sistemas de recomendación	32
1.5.1 Definición.....	32
1.5.2 Usos	32
1.5.3 Características.....	33
1.6 Trabajos relacionados	36
1.6.1 Fab.....	36
1.6.2 SiteSeer	37
1.6.3 Recomendador PTV	37
1.6.4 Outfoxed	38
1.7 Conclusión.....	39

CAPÍTULO 2	41
ANÁLISIS DE HERRAMIENTAS DISPONIBLES	41
2.1 Técnicas para solicitar la información al usuario	42
2.1.1 Calificaciones explícitas (Explicit ratings)	42
2.1.2 Calificaciones implícitas (Implicit ratings).....	42
2.2 Herramientas de software para solicitar información al usuario .	43
2.2.1 FOAF	43
2.2.2 Marcadores	45
2.2.3 Registro de Navegación Web	46
2.2.4 Navegador Web Curious.....	47
2.3 Técnicas de extracción de información.....	49
2.3.1 Reconocimiento de nombres de entidades (named entity recognition).....	49
2.3.2 Análisis de texto (Text analytics)	49
2.3.3 Minería de texto (Text mining)	50
2.3.4 Web semántica (Semantic Web)	50
2.4 Software para la extracción de información	52
2.4.1 ClearForest.....	52
2.4.2 RapidMiner	53

2.5	Estrategias de búsqueda de información.....	54
2.5.1	Recuperación de la información	54
2.5.2	Filtrado de la información.....	58
2.5.3	Diferencias entre recuperación y filtrado de información	62
2.6	Metodologías de recomendación de información a usuarios.....	64
2.6.1	Interacción humano – recomendador	64
2.6.2	Recomendación basada en colaboración de usuarios	65
2.6.3	Recomendación basada en contenido	67
2.7	Técnicas de interoperabilidad entre aplicaciones de software	68
2.7.1	Servicios Web.....	68
2.7.2	Comunicación entre procesos.....	69
2.8	Métricas	71
2.8.1	Métricas estadísticas	71
2.8.2	Métricas de soporte de decisiones.....	77
2.9	Análisis técnico.....	84
2.9.1	Tecnologías y lenguajes usados en la creación de extensiones para Firefox	84
2.9.2	Protocolos de servicios Web	87
2.10	Conclusión.....	89

CAPÍTULO 3	90
ANÁLISIS DE LA SOLUCIÓN	90
3.1 Necesidades y requerimientos	91
3.1.1 Requerimientos funcionales	92
3.1.2 Requerimientos no funcionales.....	92
3.2 Análisis de casos de uso	93
3.2.1 Diagrama general de casos de uso	94
3.2.2 Descripción de casos de uso	95
3.3 Análisis del tipo de recomendación	100
3.4 Análisis de las alternativas y selección de la solución	102
3.5 Conclusión.....	105
CAPÍTULO 4	106
DISEÑO	106
4.1 Diseño general	107
4.2 Diseño de la recomendación	110
4.3 Diagrama de clases	112

4.4	Diagramas de interacción de objetos	115
4.5	Diseño de la base de datos	125
4.6	Conclusión.....	127
CAPÍTULO 5		129
IMPLEMENTACIÓN		129
5.1	Estándares de implementación del sistema	130
5.2	Tecnologías	131
5.3	Proceso de implementación del sistema.....	132
5.4	Requerimientos de hardware y software.....	135
5.5	Problemas de implementación.....	138
5.6	Conclusiones.....	141
CAPÍTULO 6		142
PRUEBAS		142
6.1	Esquema de pruebas	143
6.2	Análisis de resultados	146

CONCLUSIONES Y RECOMENDACIONES	149
ANEXOS	154
Anexo A	155
Anexo B	161
BIBLIOGRAFÍA.....	165

ÍNDICE DE FIGURAS

Figura 2-1 Algunas aplicaciones de administración de enlaces.....	46
Figura 2-2 Navegador Web Curious.....	48
Figura 2-3 RapidMiner.....	53
Figura 2-4 Modelo más simple para la recuperación de información...55	
Figura 2-5 Curva ROC.....	81
Figura 2-6. Curva Precisión – Revisita.....	82
Figura 2-7 Llamada a procedimientos remotos.....	89
Figura 3-1 Diagrama de casos de uso.....	91
Figura 4-1 Arquitectura del sistema.....	107
Figura 4-2 Módulos del cliente.....	109
Figura 4-3 Diagrama de clases de Recommedia.....	113
Figura 4-4 Diagrama de secuencia: obtener recomendaciones cliente.....	116
Figura 4-5 Diagrama de secuencia: obtener recomendaciones servidor.....	118
Figura 4-6 Diagrama de secuencia: enviar calificación cliente.....	120
Figura 4-7 Diagrama de secuencia: enviar calificación servidor.....	121
Figura 4-8 Diagrama de secuencia: dar retroalimentación cliente.....	122
Figura 4-9 Diagrama de secuencia: dar retroalimentación servidor.....	124
Figura 4-10 Diseño base de datos Recommedia.....	126

Figura 5-1 Capas utilizadas en la implementación del proyecto de tesis.....	132
Figura 5-2 Ventana de configuración de Slogger.....	136
Figura 5-3 Sección Available Profiles de Slogger.....	137
Figura 5-4 Ventana de edición de perfiles de Slogger.....	138

ÍNDICE DE TABLAS

Tabla 2-1 Ejemplo de calificaciones y predicciones de recomendaciones.....	72
Tabla 2-2 Cálculo de la función de similitud del Coseno para el ejemplo de los vectores P y N.....	76
Tabla 2-3 Evento verdadero versus diagnóstico – La base de las curvas ROC (de Swets 1988).....	79
Tabla 3-1 Cuadro comparativo de los lenguajes de programación más importantes para desarrollar aplicaciones.....	104
Tabla 4-1 Rangos positivos y negativos.....	111

INTRODUCCIÓN

El auge de la computación y el uso masivo de la Internet han ocasionado un crecimiento explosivo de la información. Las personas pueden escoger desde decenas a cientos de libros, miles de vídeos y canciones, o millones de sitios Web disponibles en la Internet, teniendo a su disponibilidad miles de posibilidades; unas de mejor calidad que otras. Sin embargo, evaluar estas alternativas, toma el mismo tiempo y esfuerzo que antes.

Los seres humanos no evalúan todas las alternativas, sino aquellas que son de su interés, pero cuando deben realizar una elección sin ningún

conocimiento personal de ellas, como un acto humano natural, recurren a las opiniones o experiencias de otros. Las personas buscamos recomendaciones de quienes están familiarizados con nuestros objetos de decisión, quienes han sido de ayuda en el pasado, aquellos cuyos puntos de vista valoramos, o quienes son expertos reconocidos. Podemos también buscar recomendaciones de amigos o colegas, en revistas o periódicos, resultados de encuestas o evaluaciones de los productos. Por último podemos encontrar recomendaciones importantes en conversaciones sociales.

Los sistemas de recomendación han surgido con el incremento de los estudios en el aprendizaje computacional y las necesidades humanas originadas por el crecimiento de la World Wide Web. Estos sistemas buscan apoyar, mejorar, o automatizar el proceso de dar recomendaciones, por lo que hoy en día más personas los utilizan para facilitar el proceso de búsqueda de información de su interés.

El propósito de esta Tesis de Graduación es explorar el campo de los sistemas de recomendación. Nuestro objetivo central es diseñar e implementar un sistema de recomendación capaz de sugerir información de vídeos, música y sitios Web a los usuarios basándose en sus preferencias e historial y realizar pruebas midiendo la efectividad de la

recomendación. Además, una característica importante que se desea lograr en este sistema es que sea adaptivo, es decir, que tenga la capacidad de adaptarse en respuesta a los cambios del usuario.

Para la implementación del sistema analizaremos las metodologías existentes en el campo de la Inteligencia Artificial para recomendación de información a los usuarios así como se realizará un análisis de las métricas adecuadas para enlazar las tareas de la búsqueda de información a recomendación de información.

CAPÍTULO 1

MARCO TEÓRICO

En este capítulo se determina que la sobrecarga de información es un problema que puede ser solucionado con la utilización de un sistema de recomendación. Así mismo, se describen los principales antecedentes del problema, se consideran las motivaciones del desarrollo de este proyecto de tesis, se precisan sus limitaciones y los objetivos que se quiere lograr junto con una revisión de los trabajos relacionados.

1.1 Definición del problema

El crecimiento sorprendente de Internet ha hecho posible obtener una gran cantidad de información sobre cualquier tópico. Frecuentemente, encontramos más libros, películas y hasta programas de televisión que las personas pueden manejar efectivamente. Paradójicamente, la información disponible es tanta que los usuarios finales están sobrecargados de datos. El tiempo requerido para buscar a través de todos los ítems disponibles es demasiado para los usuarios comunes, y los lleva a dejar el “proceso de búsqueda” sin ningún resultado [2].

En el mundo real las personas reciben guías de recomendación de amigos, críticos o sugerencias en los medios de comunicación que les ayudan en el proceso de decisión, este escenario trasladado a Internet da origen a los sistemas de recomendación. Estos sistemas tratan de ser una alternativa al proceso social de recomendación, es decir, el acto habitual que todos practicamos al recurrir a las opiniones de conocidos o expertos cuando tenemos que tomar una decisión sin tener la suficiente información para ello. Estas decisiones son generalmente sencillas y se encuentran en el entorno de la vida cotidiana como por ejemplo qué enlace visitar, qué película ver o qué música escuchar [3].

Esta tesis mostrará cómo se puede utilizar un sistema de recomendación para aliviar el problema de la sobrecarga de información de los usuarios de la World Wide Web.

1.2 Antecedentes

Una vez que las computadoras fueron inventadas, los seres humanos nos hemos preguntado si éstas podrían ser mejoradas para aprender. Si pudiéramos programarlas para mejorar automáticamente con la experiencia el impacto podría ser impresionante. Un exitoso entendimiento de cómo hacer que las computadoras aprendan puede abrir nuevos usos en campos variados y nuevos niveles de competencia y personalización. Y un entendimiento detallado de los algoritmos de procesamiento de información para el aprendizaje computacional puede liderar un entendimiento mejor de las habilidades humanas de aprendizaje [1].

Aún no conocemos la manera de cómo hacer que las computadoras aprendan casi tan bien como lo hacen las personas. Sin embargo, han sido inventados algoritmos que son efectivos para ciertos tipos de actividades de aprendizaje. Para problemas como reconocimiento facial o del habla los algoritmos basados en aprendizaje por computadora se

desenvuelven mejor que otros logros. En el campo conocido como minería de datos, los algoritmos de aprendizaje computacional han sido usados para descubrir conocimientos valiosos de grandes bases de datos comerciales. Finalmente, en el campo de la inteligencia artificial, exitosos problemas de búsqueda, recuperación y filtrado de información han sido resueltos utilizando aprendizaje computacional [1].

En los últimos veinte años, una solución al problema de la sobrecarga de información fue provista por las ciencias de Recuperación de la Información [2]. Sin embargo, los sistemas de recomendación han nacido como un paso más allá en la dirección de ayudar a los usuarios, difiriendo de las herramientas de Recuperación de Información clásicas en que tratan de aprender más sobre los usuarios, procuran tratar cada tipo de ítem disponible y tratan de anticiparse a las necesidades de los usuarios pudiendo ser usadas como herramientas de decisión.

1.3 Justificación

Para enero del 2005 se estimaba que el tamaño de la Web indexable [12] llegaba al menos a 11.5 billones de páginas [10]. En la actualidad, según el sitio WorldWideWebSize.com [13] la Web indexable llega al menos a 25.53 billones de páginas. El increíble crecimiento del universo

digital parece indicar que los individuos estarán enfrentando la explosión de la información en una escala sin precedentes. El costo de no enfrentar la avalancha de información puede no ser visible inmediatamente, no obstante, en encuestas a compañías estadounidenses, se encontró que los trabajadores emplean 9.6 horas de su tiempo buscando información y 9.5 analizando información [14].

Como el número de sitios Web, música, vídeos y demás continúa creciendo de manera exponencial, se está haciendo más difícil sobrellevarlo por lo que en los últimos años han emergido los sistemas de recomendación para ayudar a las personas a encontrar información relevante. Esta tesis se origina del propósito de ayudar a los usuarios a encontrar información de su interés, anticipándose a sus necesidades.

1.4 Alcance y objetivos

1.4.1 Alcance

A continuación detallaremos el alcance y los límites del sistema que proponemos con el desarrollo de esta tesis, para poder cubrir las necesidades de los usuarios de la World Wide Web.

- El sistema funcionará como una adición al navegador Mozilla Firefox y será compatible con las versiones 2.X.X.
- El sistema proveerá recomendaciones de sitios Web, música y vídeos en Internet.

1.4.2 Objetivos

Durante el desarrollo de este proyecto de tesis perseguiremos los siguientes objetivos:

- Analizar las metodologías existentes en el campo de la inteligencia artificial para recomendación de información a los usuarios.
- Analizar métricas adecuadas para enlazar las tareas de búsqueda de información a recomendación de información.
- Diseñar un sistema de recomendación que sugiera información (vídeos, música y sitios Web) al usuario conociendo sus preferencias e historial, basado en el modelo Human Recommender Interaction (HRI). Este sistema será adaptivo, es decir, que sea cambiante en respuesta a los cambios en los perfiles de los usuarios.
- Implementar un sistema de recomendación basado en el diseño propuesto. Este sistema podrá instalarse como una adición a un navegador Web.
- Medir la efectividad de la recomendación que el sistema realiza.

1.5 Sistemas de recomendación

1.5.1 Definición

Un recomendador es un sistema capaz de predecir si a un usuario le agrada o no cierto ítem. Usando el término 'ítem' nos referimos a cada objeto, medio o información que un usuario podría acceder. Un ítem puede ser un libro, una película, un restaurante o una página Web. Potencialmente cada objeto que interactúa con el usuario puede ser recomendado. No hay restricciones en que es un ítem, y por supuesto, no hay límites sobre su recomendación [2].

1.5.2 Usos

El surgimiento de la Web 2.0 fomentó el uso de los sistemas de recomendación debido al crecimiento de comunidades basadas en el Web, redes sociales y arquitecturas de participación en donde los usuarios pueden colaborar y compartir experiencias. Actualmente, estos

sistemas son utilizados para recomendar películas, libros, música, amigos, sitios de entretenimiento, etc.

1.5.3 Características

Presentamos un breve resumen de todas las características que pensamos son importantes en un sistema de recomendación [2].

Cambios en los intereses de los usuarios (sistema adaptivo)

El proceso de recomendación actúa en un ambiente dinámico, en el cual dos variables pueden cambiar frecuentemente: La naturaleza de la información y los intereses del usuario en diferentes categorías [2].

Perfiles diferentes en medios y tópicos diferentes

Esta característica indica que el perfil especializado de un usuario en un dominio específico, por ejemplo, libros, no se desempeñará de la misma manera en un dominio diferente. Este hecho hace necesario tener diferentes perfiles para los diferentes dominios, según el alcance de la aplicación [2].

Capacidad de dar una explicación

Para que el usuario pueda entender de mejor manera como trabaja el sistema de recomendación, y pueda actuar de mejor manera con dicho sistema, es un requerimiento importante dar una explicación simple sobre las predicciones [2].

Capacidad de dar un coeficiente de confianza

Un coeficiente de confianza es un indicador de la seguridad que tiene el sistema de recomendación sobre las predicciones que realiza. Esta característica es importante porque permite a los usuarios tener un mejor juicio sobre los resultados de la recomendación [2].

Retroalimentación explícita e implícita

Cada sistema de recomendación debería ser capaz de aceptar retroalimentación de los usuarios, en ambas formas, explícita e implícita [2].

Función de similitud para usuarios

Cada sistema de recomendación debería implementar una función que recibe dos usuarios como entrada, y retorna que tan similares son entre ellos. Esta función de similitud es necesaria si el sistema implementa un algoritmo de filtrado colaborativo [2].

Función de distancia para usuarios

Se relaciona fuertemente con la función de similitud, una función de distancia es útil en el filtrado colaborativo. Difiere de la función de similitud porque esta función debería ser una distancia, con todas las propiedades relacionadas [2].

Permitir estereotipos

Es importante que el sistema pueda usar estereotipos de usuarios, principalmente en las etapas tempranas de uso. Como algunos sistemas requieren un periodo largo de entrenamiento, es útil usar estereotipos de usuarios para realizar sugerencias, pues, de otra manera, algunos usuarios dejarán de utilizar el sistema antes de que trabaje de forma correcta [2].

Capacidad de realizar proposiciones innovadoras

Los sistemas de recomendación tienden a sugerir siempre ítems que se ajustan muy bien con los gustos del usuario. Esta es su definición. Pero el usuario puede sentirse aburrido de recibir el mismo tipo de ítems. Por tal razón los sistemas de recomendación deberían ser capaces de recomendar también ítems que están un poco lejos de los gustos de los usuarios. Los ítems innovadores son importantes para la satisfacción del usuario porque añaden un valor adicional a los recomendadores estándares [2].

1.6 Trabajos relacionados

Existen diversos sistemas de recomendación realizados en los últimos años. Hemos escogido cuatro sistemas de recomendación representativos, debido a la forma en que cada uno realiza el proceso de recomendación, los cuales describiremos a continuación:

1.6.1 Fab

Fab es un sistema de recomendación de URLs para la World Wide Web. El sistema Fab utiliza los filtrados colaborativo y basado en el contenido combinados. El usuario objetivo se registra en el sistema, el

cual crea una página conteniendo las URLs referenciando a las páginas más recomendadas para el usuario [15].

1.6.2 SiteSeer

SiteSeer es un sistema de recomendación de páginas Web que usa marcadores individuales y la organización de marcadores dentro de carpetas para predecir y recomendar páginas relevantes. SiteSeer utiliza los marcadores de cada usuario como una declaración implícita de interés en el contenido, y el comportamiento de agrupación del usuario (como la colocación de los temas en las carpetas) como un indicador de la coherencia semántica o agrupamientos relevantes entre temas.

Además, SiteSeer trata a las carpetas como un sistema de clasificación personal que les permite adaptar las recomendaciones en las clases definidas por los usuarios [16].

1.6.3 Recomendador PTV

PTV es un sistema de recomendación diseñado para hacer sugerencias de programas de televisión, basadas en las preferencias de cada usuario. Los perfiles de PTV contienen listas de programas de televisión calificados de manera positiva y negativa.

Podemos ser capaces de encontrar coincidencias en los perfiles descubriendo relaciones entre programas más allá de una simple superposición directa y así hacer más efectiva la recomendación [17].

1.6.4 Outfoxed

Outfoxed es la implementación de la tesis de maestría de Stan James, en la Universidad de Osnabrück, Alemania [62]. En pocas palabras Outfoxed usa la red de amigos confiables para determinar que es bueno, malo y peligroso en Internet. Esto se hace añadiendo un complemento al explorador Firefox. La codificación empezó el 27 de Diciembre del 2004.

La idea esencial de Outfoxed es que los usuarios hagan decisiones basadas principalmente en la opinión de gente en quien ellos confían. La persona promedio tiene un conjunto de expertos a quienes consultar en diferentes áreas: computación, mecánica, moda, economía. Si las

opiniones de estos expertos pueden ser recolectadas, son increíblemente útiles: es la meta data (datos acerca de otros datos) que da los más inteligentes filtrado y clasificación de la información en Internet.

Outfoxed lee y publica evaluaciones de confianza como RSS, con un par de etiquetas. Esta disposición podría decir que xyz.com es bueno, mientras que zyx.com es malo, y xxx.com es peligroso. Pero la clave de todo es que en estos ficheros RSS podemos hablar de otros archivos RSS. Así que un archivo puede decir que el archivo en xyz.com/mary.xml contiene información digna de confianza, mientras que el archivo en abc.com/bob.xml debe pasarse por alto.

Como cualquier lector de RSS, Outfoxed lee periódicamente estos archivos descargados de la Web, y mantiene una base de datos local de todos los informes [6].

1.7 Conclusión

En este capítulo analizamos el problema relacionado con la sobrecarga de información y los antecedentes del mismo. De igual manera justificamos la realización del presente proyecto de tesis indicando sus alcances y objetivos.

Además revisamos la definición, usos y características de los sistemas de recomendación y como estos pueden resolver el problema planteado.

Finalmente hicimos un recuento de los principales trabajos relacionados y que pueden servirnos de apoyo y guía.

CAPÍTULO 2

ANÁLISIS DE HERRAMIENTAS DISPONIBLES

En este capítulo se explican las tecnologías utilizadas para solicitar, extraer, buscar y recomendar información. Luego, se exponen las diferentes técnicas de interoperabilidad entre aplicaciones de software y las métricas utilizadas en sistemas de recomendación. Finalmente se realiza un análisis técnico de las herramientas disponibles para la solución del problema

2.1 Técnicas para solicitar la información al usuario

Los sistemas de recomendación requieren de una interfaz de usuario que pueda determinar "inteligentemente" el interés de un usuario y utilizar esta información para hacer sugerencias. Existen dos técnicas para solicitar la información al usuario.

2.1.1 Calificaciones explícitas (Explicit ratings)

Usar calificaciones explícitas es la solución más común, en donde los usuarios dicen al sistema que opinan sobre algún objeto (ej. un CD de música) o de una parte de la información que ven (ej. un artículo en un diario electrónico). Las calificaciones explícitas son bien entendidas y bastante precisas.

Sin embargo, hacer una pausa para calificar explícitamente la información recibida puede alterar los patrones normales de navegación y lectura de los usuarios [21].

2.1.2 Calificaciones implícitas (Implicit ratings)

Este método consiste en adquirir las calificaciones por un método distinto al de obtenerlas directamente del usuario, al igual que registrar las acciones del usuario en el navegador Web, acciones como el tiempo de visita en una página, abrir un enlace o borrar una recomendación.

Estos indicadores de interés implícitos evitan el trabajo al usuario de realizar las calificaciones explícitas, y cada acción del usuario en el sistema puede contribuir a una calificación implícita [21].

2.2 Herramientas de software para solicitar información al usuario

Para poder recomendar la información de manera adaptiva, un sistema debe solicitar información al usuario, de modo que éste provea calificaciones que reflejen sus gustos para cada ítem recomendado.

Para solicitar la información al usuario, sea de manera implícita o explícita, tenemos las siguientes opciones:

2.2.1 FOAF

La tecnología de la Web Semántica puede servir para poder utilizar indicadores implícitos de interés. Un formato de la Web Semántica es FOAF.

El proyecto Friend of a Friend (FOAF) se basa en crear una red de páginas que puedan ser leídas por las computadoras. Estas páginas describen personas, los enlaces entre la gente y las cosas que crean y hacen, tales como fotos, calendarios o blogs.

FOAF es una tecnología simple que ha sido diseñada para permitir la integración de datos a través de una variedad de aplicaciones, sitios y servicios Web, y sistemas de software. Si las personas publican la información en el formato de documento FOAF, las máquinas podrán hacer uso de dicha información. Los programas serán capaces de escarbar alrededor de una red de documentos diseñados para las máquinas, en lugar de humanos [36].

Las ventajas que a futuro presenta FOAF es la rápida y efectiva constitución de comunidades virtuales, búsqueda de personas con los mismos intereses, documentos escritos por otros, entre otros.

2.2.2 Marcadores

Con la introducción de las nuevas aplicaciones de software sociales, tales como blogs, wikis, feeds de noticias, redes sociales, y herramientas como marcadores, la Web está siendo centrada en el usuario.

Los marcadores son localizadores de páginas Web (Uniform Resource Locators) que pueden ser recuperados. Son una característica de todos los navegadores Web modernos, su propósito principal es catalogar y acceder fácilmente a las páginas Web que un usuario ha visitado o visitará, por nombre en lugar de dirección.

Actualmente existen algunas aplicaciones de administración de enlaces en la Web, las cuales son referidas como herramientas de marcadores sociales. En estas aplicaciones hay un par de cosas que se deben tener en cuenta: 1) software del lado del servidor dedicado específicamente a administrar enlaces con un enfoque en redes sociales, y 2) un logro abierto y no estructurado para que el usuario etiquete o clasifique dichos enlaces [39].

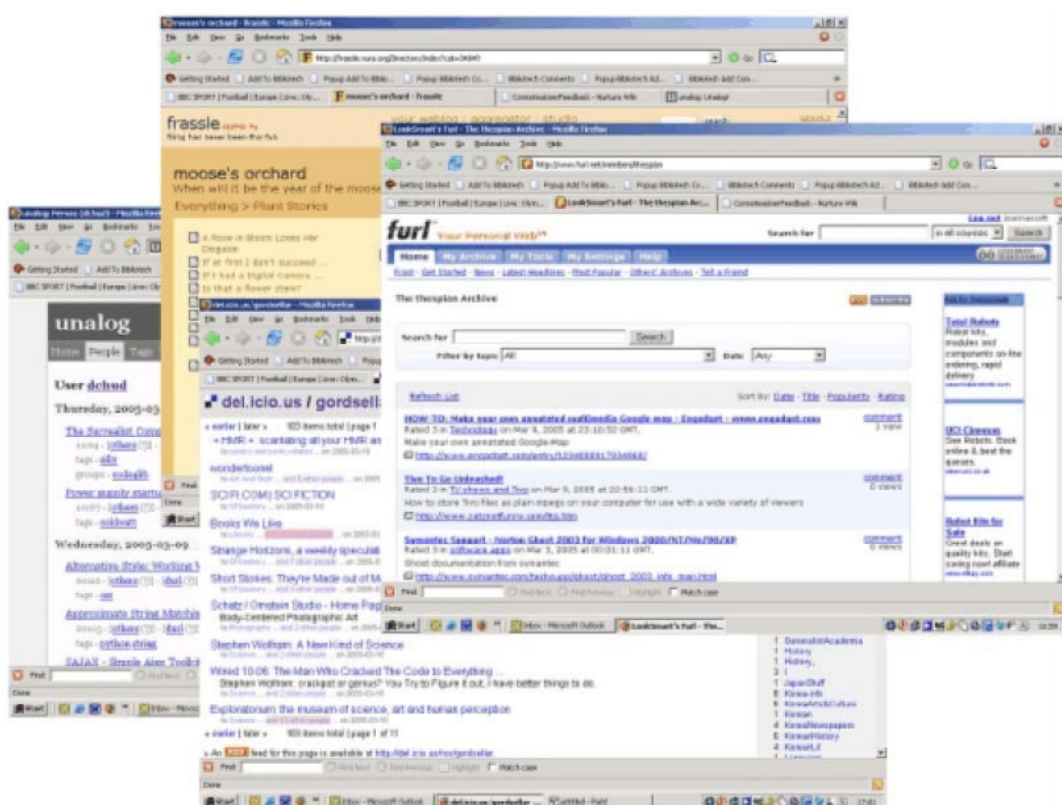


Figura 2-1 Algunas aplicaciones de administración de enlaces [39]

Los marcadores permiten a los usuarios registrar sitios de interés, y entre los más conocidos están del.icio.us, CiteULike, Flickr, Frassle, Furl, Unalog, etc. [39].

2.2.3 Registro de Navegación Web

El Registro de Navegación Web, es una extensión, cuyo nombre es Slogger, diseñada para el navegador Web Mozilla Firefox. Es una

herramienta muy flexible que sirve para crear un registro completo del historial de navegación ya que puede guardar las páginas que el usuario visita en el disco duro y crear un historial con un formato personalizado, como HTML, RDF o XML, también se puede usar hojas de estilo de cascada o programas externos para ordenar los datos según las necesidades del usuario [38].

2.2.4 Navegador Web Curious

El navegador Web Curious fue desarrollado como parte del trabajo de investigación en el estudio de Indicadores Implícitos de Interés del Instituto Politécnico de Worcester [21], realizado para registrar las acciones de los usuarios como calificaciones implícitas y las calificaciones explícitas de una página. Las acciones incluyen clics y movimientos del mouse, scrolling y tiempo transcurrido. En este estudio, el navegador fue usado por más de 70 personas que visitaron más de 2500 páginas Web [21].



Figura 2-2 Navegador Web Curious [21]

Usando los datos recolectados por el navegador, las calificaciones implícitas individuales y algunas combinaciones de éstas fueron analizadas y comparadas con las calificaciones explícitas. Los investigadores encontraron que el tiempo transcurrido en una página, la cantidad de scrolling en una página y la combinación de tiempo y scrolling tenían una fuerte correlación con el interés explícito, mientras que los valores individuales de scrolling y clicks del mouse eran inefectivos en predecir interés explícito [21].

2.3 Técnicas de extracción de información

2.3.1 Reconocimiento de nombres de entidades (named entity recognition)

El reconocimiento de nombres de entidades es una tarea en la cual nombres de personas, nombres de lugares, nombres de organizaciones, cantidades monetarias, tiempo y expresiones de porcentaje son reconocidos en un documento de texto.

Esta tarea es una técnica básica e importante para extracción de información y sistemas de pregunta-respuesta [19].

2.3.2 Análisis de texto (Text analytics)

El análisis de texto es un proceso de extracción de información cuyo objetivo es extraer automáticamente la información semi-estructurada o no estructurada de los documentos de texto.

Una aplicación típica es escanear un conjunto de documentos escritos en un lenguaje natural y llenar una base de datos o índice de búsqueda con la información extraída [19].

2.3.3 Minería de texto (Text mining)

La minería de texto puede definirse en términos generales como un proceso de conocimiento intensivo en el que un usuario interactúa con una colección de documentos en el tiempo mediante el uso de una serie de herramientas de análisis. De una manera análoga a la minería de datos, la minería de texto trata de extraer información útil de las fuentes de datos a través de la identificación y exploración de patrones interesantes.

En el caso de la minería de texto, sin embargo, las fuentes de datos son colecciones de documentos, y entre los registros de la base de datos no se encuentran interesantes patrones, sino en el texto no estructurado de datos en los documentos de estas colecciones [20].

2.3.4 Web semántica (Semantic Web)

“Para una computadora, la Web es un mundo delgado y aburrido, sin significado. Esto es una lástima, ya que en realidad los documentos en la Web describen objetos reales y conceptos imaginarios, y

proporcionan relaciones particulares entre ellos. Por ejemplo, un documento puede describir una persona. El título del documento a una casa y la relación entre el propietario con dicha persona. Añadir semántica a la Web involucra dos cosas: permitir que los documentos contengan información en formas legibles por las máquinas, y permitir que los enlaces sean creados con valores de relación. Sólo cuando tengamos este nivel extra de semántica podremos ser capaces de usar el poder de las computadoras para que nos ayuden a explotar la información a un nivel mayor que nuestro entendimiento.” [37]. De aquí la importancia de analizar el concepto y las características de la Web semántica en el estudio de los sistemas de recomendación.

La Web semántica es una Web de datos. Hay mucha información que usamos todos los días, y no es parte de la Web. Yo puedo ver mis estados de cuenta bancarios en la Web, puedo ver mis fotografías, y puedo ver mis compromisos en un calendario. Pero, ¿puedo ver mis fotos en un calendario para ver que estaba haciendo cuando las tomé? ¿Puedo ver mis estados de cuenta del banco en un calendario? ¿Por qué no? Porque no tenemos una Web de datos. Porque los datos son controlados por aplicaciones y cada aplicación los mantiene para sí misma.

La Web semántica trata acerca de dos cosas. La primera encapsula los formatos comunes para la integración y combinación de los datos procedentes de diversas fuentes. Y la segunda se centra en un lenguaje para recordar como los datos se relacionan a los objetos del mundo real. Esto permite a una persona, o a una máquina, empezar en una base de datos y luego moverse a través de una interminable serie de bases de datos que no están conectadas por cables pero están relacionadas con algo en común [22].

2.4 Software para la extracción de información

Existen múltiples técnicas para extraer información. Las más relevantes se nombraron en la sección 2.3. Luego del análisis y búsqueda de herramientas de software que utilicen las técnicas descritas en dicha sección, hemos encontrado dos herramientas relevantes que ofrecen versiones de uso libre lo cual es útil para el desarrollo de nuestra tesis de graduación. En esta sección se analizarán dichas herramientas.

2.4.1 ClearForest

ClearForest es una herramienta comercial para análisis de textos basado en entidades claves y hechos, compuesta de una plataforma de

etiquetas y extracción, una plataforma analítica y un ambiente de desarrollo [5].

Un subconjunto de las capacidades de ClearForest está disponible a través de servicios Web bajo el nombre de Semantic Web Services (SWS) que convierten el texto en información que se puede utilizar mediante formato XML; y la oportunidad de descargar una extensión para el explorador Web Mozilla Firefox denominada Gnosis [5].

2.4.2 RapidMiner

RapidMiner es una solución open – source para minería de datos que combina tecnologías líderes y un rango funcional.



Figura 2-3 RapidMiner [40]

Más de 400 operadores de minería de datos pueden ser usados. La configuración es descrita por archivos XML los cuales pueden ser fácilmente creados con una interfaz grafica de usuario. Este lenguaje de

scripting convierte a RapidMiner en un ambiente de desarrollo integrado (IDE) para aprendizaje computacional y minería de datos [40].

2.5 Estrategias de búsqueda de información

Las principales estrategias de búsqueda analítica de información son recuperación de la información y filtrado de la información [24, 25].

2.5.1 Recuperación de la información

La recuperación de la información (IR, Information Retrieval) es la ciencia que concierne los procesos relacionados a la representación, almacenamiento, búsqueda y hallazgo de información que es relevante a un requerimiento deseado por un usuario humano [24].

Tradicionalmente, la información toma la forma de texto, lo cual implica que IR es sinónimo de recuperación de documentos o texto [24]. Asume también tener una base constante de ítems y ayuda a los usuarios con intereses cambiantes [2].

El incremento enorme en la cantidad de texto en línea disponible, y la demanda de acceso a diferentes tipos de información han liderado un renovado interés en un rango amplio de áreas relacionadas con la recuperación de información que van desde recuperación de documentos, responder preguntas, detección y rastreo de tópicos, recuperación de multimedios (por ejemplo, imágenes, vídeos y música), ingeniería de software, informática médica y biológica, estructuración de texto y minería de texto [25].

El modelo más simple para la recuperación de información se muestra en la figura 2-4. A la izquierda se representa la información potencial, por ejemplo, términos indexados, estructuras gráficas o códigos de categorías así como datos formales. A la derecha, un requerimiento de información se representa por un query, en lenguaje natural o en un lenguaje artificial de sentencias. En el centro, una función de unión compara las representaciones con el query y recupera entidades de texto, por ejemplo documentos o partes de documentos, que proveen la información que el usuario solicita [24].



Figura 2-4 Modelo más simple para la recuperación de información [24]

Los sistemas automatizados de recuperación de información son usados para reducir la sobrecarga de información. Algunas universidades y bibliotecas públicas usan sistemas IR para proveer acceso a libros, publicaciones y otros documentos. Los motores de búsqueda como Google, Yahoo! Search o Live Search son las aplicaciones más visibles de recuperación de información [24].

2.5.1.1 Problemas

Entre los desafíos que las ciencias de recuperación de información enfrentan a largo plazo están: el acceso global a la información y la recuperación contextual [25].

2.5.1.1.1 Acceso global a la información

Con el acceso global a la información se busca satisfacer las necesidades humanas de información a través de interacción natural, eficiente con un sistema automatizado que tome ventaja de los datos estructurados y no estructurados a nivel mundial en cualquier lenguaje [25].

La información en la World Wide Web se almacena en una tasa fenomenal y en cualquier lenguaje. Un gran reto para IR es desarrollar sistemas distribuidos con recuperación multi – lenguaje. Estos sistemas podrían tomar como entrada una necesidad de información, codificada en cualquier lenguaje [25].

2.5.1.1.2 Recuperación contextual

Este problema se basa en combinar tecnologías de búsqueda y conocimiento acerca del contexto de queries y usuarios en un solo marco de trabajo con el objeto de proveer la respuesta más apropiada para las necesidades de información de un usuario [25].

Un ejemplo de recuperación contextual podría ser un sistema conocedor del contexto. Con este sistema, si un usuario ingresa un query como “Taj Mahal”, y si el usuario está en una computadora de escritorio y ha estado planeando un viaje a la India (leyendo correos del viaje, examinando páginas de agencias de viajes, ingresando entradas en un diario), entonces el sistema conocerá este contexto y estará más inclinado a devolver imágenes y vídeos del mausoleo indio, en lugar de música de la banda de jazz con el mismo nombre.

2.5.2 Filtrado de la información

El filtrado de información (IF, Information Filtering) está principalmente relacionado con la selección o eliminación de documentos proveniente de un flujo de datos dinámicos. Normalmente se procesa un documento a la vez, pero una gran cantidad de diferentes consultas son aplicadas al mismo. Una vez procesados los documentos son entregados a los usuarios cuyos perfiles concuerdan [23].

Los sistemas de filtrado de información pueden ayudar a los usuarios eliminando la información irrelevante y mostrando sólo la información relevante [2].

Una variedad de técnicas han sido propuestas para filtrar información, incluyendo tales como demográficas, colaborativas, basadas en utilidad, contenido, y conocimiento. Para mejorar su desempeño, estos métodos han sido combinados en filtros híbridos [2].

2.5.2.1 Filtro demográfico

El filtrado demográfico agrupa a los usuarios en clases dependiendo de sus características demográficas y se asume que los usuarios

pertenecientes a un grupo tienen preferencias similares. Generaliza los intereses de los usuarios, así que el sistema recomienda los mismos objetos a usuarios con el perfil demográfico similar [11].

Este tipo de filtrado es útil en sistemas de recomendación de música pues se ha demostrado que las características demográficas tales como género, condición social y edad afectan los gustos musicales de los usuarios.

2.5.2.1.1 Problemas

No posee mecanismos para adaptarse a los cambios en las preferencias individuales de los usuarios. Sin embargo el filtrado demográfico puede ser muy poderoso combinado con otra técnica [29].

Otro problema que se evidencia al utilizar este filtro es que se requiere mucho esfuerzo para obtener la información demográfica de los usuarios ya que puede haber resistencia de su parte en proporcionar dicha información por la tendencia a pensar en que se viola la privacidad.

2.5.2.2 Filtro basado en utilidad

El filtro basado en utilidad no construye generalizaciones a largo plazo, en su lugar, compara la necesidad del usuario con el conjunto de opciones disponible, mediante una función de la utilidad de cada objeto para el usuario; la cual sería su perfil. Luego emplea técnicas de satisfacción de restricciones para escoger la mejor opción. [8] Como es de suponer, el problema central es como crear una función de utilidad para cada usuario [11].

2.5.2.2.1 Problemas

Requiere que el sistema construya una función de utilidad completa sobre todas las características de los objetos en consideración. Un beneficio de ese logro es que puede incorporar algunos factores diferentes que contribuyen al valor de un ítem [11].

2.5.2.3 Filtro basado en el conocimiento

Se caracteriza por disponer de información sobre cómo un ítem satisface una necesidad del usuario y establece relación entre una necesidad y una posible recomendación [11]. Utiliza reglas, patrones o conexiones entre los ítems para generar sugerencias; por ejemplo,

cuando alguien compra una lámpara, el sistema le sugiere que también compre algunos bombillos [31].

2.5.2.3.1 Problemas

El principal problema de este tipo de filtrado es que necesita adquisición del conocimiento:

- **Conocimiento del catálogo:** Conocimiento sobre los objetos siendo recomendados y sus características.
- **Conocimiento funcional:** El sistema debe ser capaz de cotejar entre las necesidades del usuario y el objeto que puede satisfacer dichas necesidades.
- **Conocimiento del usuario:** Para proveer buenas recomendaciones, el sistema debe tener cierto conocimiento sobre el usuario. Lo cual puede parecer que el filtro toma la forma de información general demográfica o información específica sobre la necesidad para la que se busca la recomendación.

De estos tipos de conocimiento, el último es el problema más desafiante de resolver, puesto que el gusto de dos usuarios por determinado ítem se puede deber a razones completamente distintas. Por lo tanto, estas

razones diferentes originan implicaciones diferentes para las siguientes recomendaciones para ambos usuarios y también como la información de aquellos usuarios deberá ser usada para influenciar las recomendaciones para otros [30]. Obtener la razón o la necesidad que motiva a un usuario a preferir un ítem es muy difícil para un sistema de recomendación debido a que hay gran variedad de razones específicas que influyen en el proceso de decisión de una persona y el sistema no podrá cubrir todos los motivos posibles.

2.5.2.4 Filtro híbrido

Los sistemas híbridos combinan dos o más técnicas de filtrado para ganar mejor desempeño con menos inconvenientes que alguna técnica individual. Comúnmente, el filtrado colaborativo se combina con otra técnica.

2.5.3 Diferencias entre recuperación y filtrado de información

En un nivel abstracto las entidades y procesos relevantes al filtrado de información (IF, Information Filtering) son casi idénticos a aquellos relevantes a la recuperación de información (IR, Information Retrieval).

Las principales diferencias pueden ser expresadas en unos cuantos puntos sobresalientes:

- IR se enfoca en retornar documentos relevantes de una base estática, mientras IF se enfoca en seleccionar documentos relevantes de una base cambiante constantemente [31].
- Mientras IR es típicamente relacionada con usuarios individuales del sistema, por una persona con un objetivo y un query, IF se relaciona con usos repetidos del sistema, por una persona o personas con objetivos a largo plazo o intereses [28].
- IR se relaciona con la colección y organización de textos, en cambio, el filtrado se relaciona con la distribución de textos a grupos o individuos [28].
- Aspectos ambientales. La mayor diferencia parece ser que el filtrado está altamente involucrado, en algunas situaciones, con cuestiones de privacidad. IR, por una variedad de razones, casi no ha prestado atención a esta clase de problema [28].

Debido a las diferencias relativamente pequeñas entre ellas, las ciencias de recuperación y filtrado de la información son consideradas “dos lados de la misma moneda” puesto que sus objetivos son esencialmente equivalentes: trabajan para ayudar a las personas a obtener la información necesaria para realizar sus actividades.

2.6 Metodologías de recomendación de información a usuarios

Históricamente, al hablar de metodologías de recomendación de información a usuarios, las investigaciones [7, 29, 31] se han enfocado en realizar recomendaciones más precisas, con la asunción implícita que *más preciso es igual a que gusten más y sean más útiles*. Pero primero es necesario determinar qué clase de recomendaciones son buenas y útiles a los usuarios para finalmente volver a pensar en los recomendadores desde una perspectiva centrada en el usuario, la teoría de la interacción humano – recomendador es una manera de incorporar este conocimiento en los sistemas de recomendación.

2.6.1 Interacción humano – recomendador

La interacción humano – recomendador (HRI, Human – Recommender Interaction) es un marco de trabajo y metodología para analizar las tareas de usuarios y algoritmos de filtrado y recomendación con el objetivo final de generar listas útiles de recomendación. Fue desarrollado reexaminando el proceso de recomendación desde la

perspectiva del usuario final: HRI categoriza aspectos de las interacciones que un usuario tiene con un recomendador basado en las experiencias y expectativas del usuario desarrolladas sobre un periodo de tiempo. Es un lenguaje para describir las clases de recomendaciones que podrían ayudar a resolver la necesidad de información de un usuario [7].

2.6.2 Recomendación basada en colaboración de usuarios

La idea básica de los algoritmos basados en filtrado colaborativo es proveer recomendaciones de ítems o predicciones basados en las opiniones de otros usuarios con pensamientos similares. Las opiniones de los usuarios pueden ser obtenidas explícitamente de los usuarios o usando alguna medida implícita [27]. El sistema busca usuarios con intereses similares a los de un usuario dado y le recomienda los objetos preferidos por dichos usuarios. En lugar de calcular la semejanza entre los objetos, el sistema calcula la semejanza entre los usuarios. En la visión colaborativa, no hay análisis del contenido de los objetos. A cada objeto se le asigna un único identificador y una calificación definida por el usuario. La semejanza entre los usuarios está basada en la comparación de las calificaciones que los usuarios dan a los mismos objetos [29].

Los sistemas de filtrado colaborativo han sido muy exitosos en el pasado, pero su extenso uso ha revelado retos potenciales como:

- **Problema del primer clasificador:** Cuando un sistema colaborativo es creado, contiene muchos ítems, pocos usuarios y ninguna clasificación. Sin clasificaciones, el sistema no puede generar recomendaciones y los usuarios no ven beneficio alguno. Sin usuarios, no hay manera de que las nuevas clasificaciones sean ingresadas en el sistema. Este inconveniente se conoce como *problema del primer clasificador* [31].
- **Problema del nuevo usuario:** Otro problema común es aquel del *nuevo usuario* ya que antes de que un usuario pueda tomar ventaja de un sistema de filtrado colaborativo, primero debe proveer sus opiniones. El usuario tiene que confiar que las recomendaciones que el sistema genere valdrán el esfuerzo de ingresar sus opiniones. Esto es difícil, debido a que esta confianza es necesaria antes de que el usuario empiece a utilizar el sistema [31].
- **Problema de esparcimiento:** En algunos dominios, un usuario tiende a clasificar solo un pequeño porcentaje de los

ítems disponibles, lo cual puede hacer difícil encontrar concordancias entre individuos [31].

2.6.3 Recomendación basada en contenido

En este esquema se escogen los objetos basándose en las características de sus contenidos [4]. El filtro busca objetos similares a aquellos que el usuario prefiere basado en una comparación del contenido: información y meta – data capturada de los ítems en el dominio, como por ejemplo, el texto de los libros comprados con anterioridad [31].

El filtrado basado en el contenido tiene algunas desventajas, principalmente en la captura de diferentes aspectos del contenido, por ejemplo, música, vídeos e imágenes. Pero también para el caso de textos, las representaciones que se hacen del documento, capturan únicamente ciertos aspectos del contenido, que resultan en un pobre desempeño de los sistemas. Además de los problemas de representación, estos sistemas tienden a aprender de forma tal que ellos recomiendan objetos similares a los objetos ya vistos en usos anteriores del sistema [29].

2.7 Técnicas de interoperabilidad entre aplicaciones de software

2.7.1 Servicios Web

La W3C define a un servicio Web como: “Un sistema de software diseñado para soportar interacción interoperable máquina a máquina sobre una red. Tiene una interfaz descrita en un formato procesable por computadora (específicamente WSDL). Otros sistemas interactúan con el servicio Web de la manera establecida por su descripción usando mensajes SOAP, transportados típicamente usando HTTP con serialización XML en conjunción con otros estándares Web relacionados.” [32]

Los Servicios Web permiten la comunicación entre aplicaciones o componentes de aplicaciones de forma estándar a través de protocolos comunes y de manera independiente al lenguaje de programación, plataforma de implantación, formato de presentación o sistema operativo. Un Servicio Web es un contenedor que encapsula funciones específicas y hace que estas funciones puedan ser utilizadas en otros servidores.

2.7.1.1 Aplicaciones Web Híbridas

Una aplicación Web híbrida (Mashup o remezcla) es un sitio Web o aplicación Web que usa contenido de más de una fuente para crear un nuevo servicio completo. Combina dos o más flujos separados de datos para crear contenido original [33].

2.7.2 Comunicación entre procesos

A pesar de que muchas tareas pueden realizarse en procesos independientes, la mayoría de ellas necesitan de la intervención de más de un proceso. Para que esta cooperación entre procesos se lleve a cabo se requiere de algún tipo de comunicación entre ellos [55, 56].

La comunicación entre procesos, en inglés IPC (Interprocess Communication) es una función básica de los Sistemas operativos. Los procesos pueden comunicarse entre sí a través de compartir espacios de memoria, ya sean variables compartidas o buffers, o a través de las herramientas provistas por las rutinas de IPC [55, 56].

Los mecanismos de comunicación entre procesos habilitan a éstos para que puedan intercambiar datos y sincronizarse, comúnmente a través de un sistema de bajo nivel de paso de mensajes que ofrece la red subyacente [55, 56].

En la comunicación entre procesos se presentan dos escenarios:

- Que los procesos se ejecuten en la misma máquina.
- Que los procesos se ejecuten en máquinas distintas.

La comunicación se establece siguiendo una serie de reglas (protocolos de comunicación). Los protocolos desarrollados para internet son los mayormente usados: IP (capa de red), protocolo de control de transmisión (capa de transporte) y protocolo de transferencia de archivos, protocolo de transferencia de hipertexto (capa de aplicación) [55, 56].

Las técnicas de IPC están divididas dentro de métodos para: paso de mensajes, sincronización, memoria compartida y llamadas de procedimientos remotos (RPC). El método de IPC usado puede variar dependiendo del ancho de banda y latencia (el tiempo desde el pedido de información y el comienzo del envío de la misma) de la comunicación entre procesos, y del tipo de datos que están siendo comunicados [55, 56].

2.8 Métricas

La investigación de sistemas de recomendación ha utilizado diferentes tipos de medidas para evaluar la calidad de un sistema de recomendación a través de la precisión, definida como la capacidad de un sistema de recomendación para ordenar los ítems según relevancia [2].

2.8.1 Métricas estadísticas

Evalúan la precisión comparando predicciones numéricas contra las clasificaciones de los usuarios para los ítems [2].

2.8.1.1 Error de media absoluta

El error de media absoluta (MAE, Mean Absolute Error) se define como el promedio del error absoluto, calculado como la diferencia entre la clasificación dada por el usuario y la predicción [2].

Teniendo el siguiente conjunto de datos:

$$\text{Conjunto de clasificaciones} = \{r_1, \dots, r_n\} \quad r_i \in \mathfrak{R}$$

Conjunto de predicciones = $\{p_1, \dots, p_n\}$ $p_i \in \mathfrak{R}$

$$\text{Error} = E = \{E_1, \dots, E_n\} = \{p_1 - r_1, \dots, p_n - r_n\} \quad [2]$$

El error de media absoluta se define como:

$$|\bar{E}| = \frac{\sum_{i=1}^n |E_i|}{N} \quad [2]$$

El objetivo al utilizar esta métrica es minimizar este error [2].

En la siguiente tabla, tenemos las clasificaciones realizadas por el usuario, en un rango de 1 a 5, donde 1 indica una calificación pésima y 5 es la mejor calificación posible. Además se incluyen las predicciones calculadas por el recomendador para cada calificación.

# Recomendación	Calificación	Predicción
1	4	3
2	5	5
3	3	4

Tabla 2-1 Ejemplo de calificaciones y predicciones de recomendaciones

Para calcular el error de media absoluta, utilizamos la fórmula descrita anteriormente:

$$|\overline{E}| = \frac{\sum_{i=1}^3 |E_i|}{3} \quad |\overline{E}| = \frac{|3-4| + |5-5| + |4-3|}{3} \quad |\overline{E}| = \frac{2}{3} \quad |\overline{E}| = 0.67$$

Este error se utiliza no sólo para medir la precisión de las recomendaciones, sino también para conocer la precisión de predicciones de diversa índole, tales como meteorológicas o de ventas [57, 58].

2.8.1.2 Error de raíz de la media cuadrada

El error de raíz de la media cuadrada (RMSE, Root Mean Squared Error) se basa en ponerles un peso desproporcionado a los errores grandes, más pesados que los errores pequeños. Esto se realiza porque algunas recomendaciones que están en 0.25 en una escala de 5 son mejores que unos cuantos que se encuentren en 3 ó 4. Su forma matemática es:

$$RMSE = \frac{\sqrt{\sum_{i=0}^n (E_i)^2}}{N} \quad [2]$$

Si tomamos los datos utilizados en el ejemplo de la sección anterior, podemos calcular el error de raíz de media cuadrada de la siguiente manera:

$$RSME = \frac{\sqrt{\sum_{i=0}^n (E_i)^2}}{N} \quad RSME = \frac{\sqrt{(3-4)^2 + (5-5)^2 + (4-3)^2}}{3}$$

$$RSME = 0.4714$$

Al igual que la métrica anterior, el objetivo es minimizar el error [2].

2.8.1.3 Correlación entre clasificaciones y predicciones

Esta métrica estadística procura establecer la correlación entre el conjunto de clasificaciones y el conjunto de predicciones. A diferencia de las métricas anteriores, el objetivo es maximizar la correlación: Mientras mayor es la correlación, mayor es la precisión del sistema. Para lograr este objetivo se pueden usar diferentes funciones, como la similitud del coseno [2].

La función de similitud del coseno viene dada por la expresión del ángulo formado por sus dos vectores asociados, donde P es el vector de las calificaciones y N es el vector de las predicciones. El resultado de la función coseno proporciona valores entre 0 y 1, en el caso de que el valor de Cos (P, N) fuese 1, indicaría que ambos casos (P, N) son idénticos y por lo tanto que los dos vectores de resultados son similares [64].

La función de similitud del coseno se muestra a continuación [63]:

$$\text{Cos}(P,N) = \frac{\sum_{i=1}^n p_i * n_i}{\sqrt{\sum_{i=1}^n p_i^2} * \sqrt{\sum_{i=1}^n n_i^2}}$$

Donde:

$$P = \begin{Bmatrix} p_1 \\ p_2 \\ \cdot \\ \cdot \\ p_n \end{Bmatrix} \quad N = \begin{Bmatrix} n_1 \\ n_2 \\ \cdot \\ \cdot \\ n_n \end{Bmatrix}$$

Si consideramos el siguiente ejemplo de dos vectores P y N, cuyos componentes se presentan a continuación: vector P = {1; 1; 1; 1; 0; 0; 0; 0} y vector N = {0; 0; 1; 1; 0; 0; 1; 1}, la tabla a continuación refleja el cálculo de la similitud [64]:

P	N	Escalar	P²	V²
1	0	0	1	0
1	0	0	1	0
1	1	1	1	1
1	1	1	1	1
0	0	0	0	0
0	0	0	0	0
0	1	0	0	1
0	1	0	0	1
	Sumas	2	4	4
	Coseno	0.5		

Tabla 2-2 Cálculo de la función de similitud del Coseno para el ejemplo de los vectores P y N [64].

En la tabla 2-2 la tercera columna es el producto escalar de los vectores P y N, la cuarta y quinta columna recogen el resultado de elevar al cuadrado los componentes de los vectores P y N. En la fila de "Sumas" se totalizan los sumatorios del producto escalar y de los cuadrados de los vectores y se calcula la función del coseno según la fórmula anterior obteniéndose un valor de 0.5 [64].

2.8.2 Métricas de soporte de decisiones

Supongamos que un modelo de predicción de películas recomienda 100 títulos y a usted le gustan 50 de ellos, y otro modelo de predicción recomienda 200 títulos y a usted le gustan 80 de ellos. ¿Cuál es el mejor modelo de predicción? Los investigadores en recuperación de la información han desarrollado métricas de soporte de decisiones para evaluar el desempeño del modelo para casos como en este ejemplo [61].

Las métricas de soporte de decisiones evalúan la efectividad de las predicciones al ayudar a un usuario a seleccionar ítems de gran calidad del conjunto [2].

2.8.2.1 Clasificación inversa

La clasificación inversa (Reversal Rate) es una medida de cómo el sistema tiene errores grandes que pueden debilitar la confianza del usuario en un sistema de recomendación. Para poder calcularla, es necesario medir dos umbrales, uno para la predicción baja (L) y otro para la predicción alta (H). La Clasificación Inversa depende de cuantos

ítems “buenos” son clasificados bajo el umbral L, y cuántos ítems “malos” son clasificados sobre el umbral H [2].

Los inversos bajos se refieren a los casos en los cuales al usuario le disgusta fuertemente un ítem, por ejemplo, cuando le da una calificación más baja que el umbral L y el sistema recomienda dicho ítem con una alta calificación de recomendación, por ejemplo, por encima del umbral H. Los inversos altos son casos en los que al usuario le gusta mucho el ítem, pero la recomendación del sistema es pobre, por ejemplo, la calificación del usuario es mayor al umbral H, y la recomendación del sistema es menor al umbral L [65].

2.8.2.2 Característica del receptor operativo

La característica del receptor operativo (ROC, Receiver Operating Characteristic) es una medida del poder de diagnóstico de un sistema de filtrado que se obtiene del área bajo la curva ROC [2].

Para entender el concepto bajo el cual se obtiene la curva ROC, nos valemos de la tabla 2-2. Un sistema de diagnóstico busca una señal particular e ignora otros eventos llamados “ruido”. El evento es considerado “positivo” o “negativo” y el diagnóstico correspondiente es

positivo o negativo. En la tabla 2-3 hay dos formas en que el diagnóstico puede estar correcto: “verdadero – positivo” y “verdadero - negativo”. Y también hay dos casos en los que el diagnóstico puede estar equivocado: “falso - positivo” y “falso – negativo” [61].

		Evento		
		Positivo	Negativo	
Diagnóstico	Positivo	Verdadero Positivo (a)	Falso Positivo (b)	a + b
	Negativo	Falso Negativo (c)	Verdadero Negativo (d)	c + d
		a + c	b + d	a + b + c + d - N

Tabla 2-3 Evento verdadero versus diagnóstico – La base de las curvas ROC (de Swets 1988) [61]

La proporción verdadera – positiva, $a/(a + c)$, y la proporción falsa – positiva, $b/(b + d)$, capturan toda la información relacionada con precisión que es relevante del modelo. Estas proporciones son llamadas usualmente proporciones de “aciertos” o “sensibilidad” y “falsas alarmas”. En adición, la proporción verdadero – negativo se llama usualmente especificidad, que es la probabilidad de que un evento seleccionado como negativo sea evaluado como negativo por el modelo. Un buen modelo de diagnóstico proveerá algunos aciertos con pocas falsas alarmas [61].

La curva ROC contiene la mayoría de la información sobre la precisión de un predictor continuo, sin embargo, a veces se desea producir medidas cuantitativas de resumen, como el área bajo la curva, (AUC). Usualmente, en una curva ROC empírica, se estima por la regla trapezoidal, que consiste en formar trapezoides usando los puntos observados como esquinas, luego computarizar las áreas de dichos trapezoides y finalmente sumarlos, lo cual puede ser un gran esfuerzo para una curva con varios umbrales. Afortunadamente, AUC se calcula también usando un par de medidas estadísticas muy conocidas que facilitan la comparación y mejoran la interpretación [59].

La probabilidad de concordancia es la primera de estas medidas. AUC también se relaciona con la estadística Wilcoxon Rank – sum [59].

Calcular el área bajo la curva ROC es más difícil de explicar y está fuera del ámbito de estudio de esta tesis. Los métodos de cálculo mencionados están disponibles como programas computarizados, tales como MedCalc y dan un estimado de AUC.

En la figura a continuación se muestra cómo cambian los niveles de sensibilidad y especificidad, dependiendo del umbral que se escoja de

la hormona tiroidea T4, que define el hipotiroidismo. El área bajo la curva ROC de la figura es de 0.86 [60].

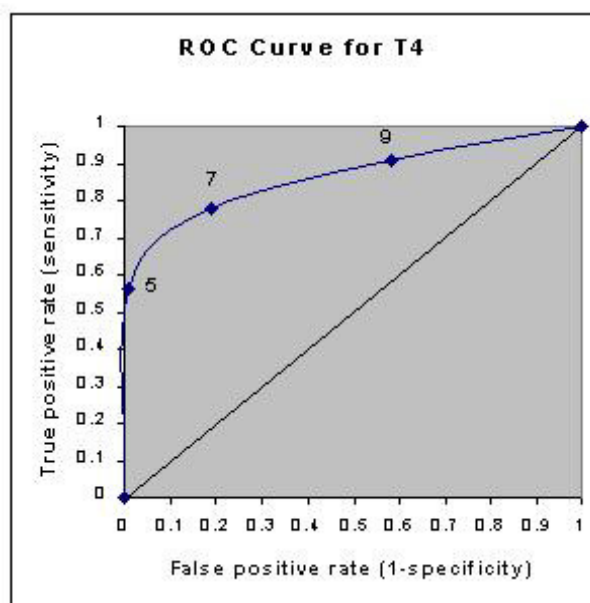


Figura 2-5 Curva ROC [60]

2.8.2.3 PRC

La métrica PRC es más simple y representa la medida de qué tanto el sistema recomienda información relevante. Matemáticamente, se define como el área bajo la curva precisión – revisita [2].

Revisita es lo mismo que la proporción verdadero – positivo o la proporción de aciertos de la curva ROC, e indica como el sistema encuentra información valiosa. La precisión es el número de verdaderos

positivos dividido para el número total de positivos diagnosticados por el modelo (Por ejemplo, $a/(a + b)$ en la Tabla 2-2) e indica cuán selectivo es el sistema. Por ejemplo, suponga que 500 es el número total de películas que se encuentran en una base de datos que le gustarán a un usuario ($a + c = 500$). Supongamos que un modelo recomienda 100 películas y que al usuario le gustan 50 de ellas ($a + b = 100$ y $a = 50$). Entonces, $b = 50$ y $c = 450$. Por lo tanto, la precisión es 0.5 ($50/100$) y la revisita es 0.1 ($=50/500$) [61].

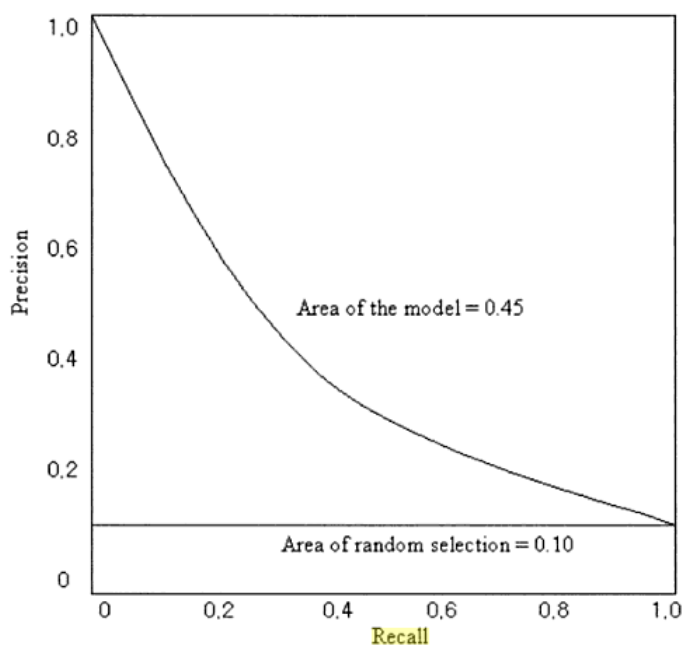


Figura 2-6. Curva Precisión – Revisita [61]

Las recomendaciones más precisas, según las métricas estándares, no siempre son aquellas más útiles a los usuarios. McNee [34] propone

una dirección centrada en el usuario para evaluar a los sistemas de recomendación basándose en las siguientes características:

- **Similitud:** Dependiendo de las intenciones del usuario, el conjunto de los ítems que aparecen en la lista afectan su satisfacción con el recomendador más que los cambios en la precisión de cada ítem en la lista. Algunos algoritmos de filtrado pueden atrapar a los usuarios en un “agujero de similitud”, solamente dando recomendaciones excepcionalmente similares, por ejemplo, una vez que un usuario ha clasificado una película de Star Trek, solamente podría recibir recomendaciones de más películas de Star Trek.
- **Serendipity:** Es la experiencia de recibir una recomendación inesperada y fortuita de un ítem. Hay un nivel de respuesta emocional asociada con esta característica que es difícil de medir en cualquier métrica.
- **Experiencias de los usuarios y expectativas:** La satisfacción del usuario no siempre se correlaciona con una gran precisión en el recomendador [35]. Los usuarios nuevos en un recomendador tienen necesidades diferentes de aquellos experimentados por lo que la elección del algoritmo

usado para usuarios nuevos afecta en gran medida su experiencia y la precisión de las recomendaciones que el sistema puede generar para él.

2.9 Análisis técnico

Como la solución propuesta por este proyecto de Tesis es la construcción de un sistema de recomendación, por lo tanto, las tecnologías y lenguajes que se analizan a continuación están relacionadas a cumplir este propósito.

2.9.1 Tecnologías y lenguajes usados en la creación de extensiones para Firefox

2.9.1.1 Lenguaje XML para interfaces de usuario

XUL (XML User Interface Language) es el lenguaje XML para interfaces de usuario de Mozilla. Permite crear potentes aplicaciones multiplataforma que pueden ejecutarse con conexión a internet o sin ella. Estas aplicaciones son fácilmente personalizables con texto alternativo, gráficos y diseños por lo que pueden ser fácilmente instalados o traducidos para diversos mercados [41].

2.9.1.2 Instalación multiplataforma

La instalación multiplataforma (XPIInstall) es una tecnología utilizada por la suite de aplicaciones de Mozilla, Thunderbird y otras aplicaciones basadas en XUL para instalar extensiones.

Un módulo instalador XPI (pronunciado “zippy” y derivado de XPIInstall) es un archivo ZIP que contiene un script instalador (llamado install.js o install.rdf) en la raíz del archivo [42].

2.9.1.3 Javascript

Javascript es un lenguaje de scripts, interpretado, multiplataforma y parcialmente orientado a objetos. Fue creado por Netscape específicamente para su uso en el desarrollo de sitios Web.

El código JavaScript puede enlazarse o añadirse a las páginas Web proporcionando un control total y dinámico sobre ellas. Además, también permite controlar -hasta cierto punto- las aplicaciones que lo ejecutan, habitualmente navegadores [43].

2.9.1.4 Modelo de objetos del documento

El modelo de objetos del documento (DOM) es un API para documentos HTML y XML. Proporciona una representación estructural del documento, permitiendo la modificación de su contenido o su

presentación visual. Esencialmente, comunica las páginas Web con los scripts o los lenguajes de programación [44].

2.9.1.5 Hojas de estilo en cascada

Hojas de estilo en cascada (Cascading Style Sheets - CSS) es el lenguaje utilizado para describir la presentación de documentos HTML o XML, esto incluye varios lenguajes basados en XML como son XHTML o SVG. Además, es usado para modificar la interfaz de usuario de aplicaciones y programas, este es el caso de los productos basados en XUL como son Firefox, Seamonkey o Thunderbird [45].

2.9.1.6 Modelo de objeto de componentes multiplataforma

El modelo de objeto de componentes multiplataforma (Cross Platform Component Object Model - XPCOM) es un marco de trabajo para escribir software modular y multiplataforma. Posee múltiples vínculos idiomáticos y descripciones IDL (lenguaje de descripción/definición de interface) con las que los programadores pueden usar sus funcionalidades propias en el marco de trabajo y conectarlo con otros componentes [46, 47].

2.9.1.7 XPConnect

XPConnect es una tecnología que permite una simple interacción entre XPCOM y Javascript.

Con XPConnect se pueden usar componentes XPCOM desde código Javascript e interactuar con objetos Javascript dentro de los componentes XPCOM [48, 49].

2.9.2 Protocolos de servicios Web

2.9.2.1 Protocolo simple de acceso a objetos

El protocolo simple de acceso a objetos (Simple Access Object Protocol – SOAP) es un protocolo ligero para el intercambio de información en un ambiente descentralizado y distribuido.

Es un protocolo basado en XML que consiste de tres partes:

- Una envoltura que define un marco de trabajo para describir lo que está en un mensaje y como procesarlo.
- Un conjunto de normas de codificación para expresar instancias de tipos de datos definidos en la aplicación.
- Una convención para representar llamadas y respuestas a procedimientos remotos [50].

2.9.2.2 Lenguaje de descripción de servicios Web

El lenguaje de descripción de servicios Web (Web Service Description Language – WSDL) es un formato XML que se utiliza para describir servicios Web.

WSDL describe la interfaz pública a los servicios Web. Está basado en XML y describe la forma de comunicación, es decir, los requisitos del protocolo y los formatos de los mensajes necesarios para interactuar con los servicios listados en su catálogo. Las operaciones y mensajes que soporta se describen en abstracto y se ligan después al protocolo concreto de red y al formato del mensaje.

Así, WSDL se usa a menudo en combinación con SOAP y XML Schema. Un programa cliente que se conecta a un servicio Web puede leer el WSDL para determinar que funciones están disponibles en el servidor. Los tipos de datos especiales se incluyen en el archivo WSDL en forma de XML Schema. El cliente puede usar SOAP para hacer la llamada a una de las funciones listadas en el WSDL [51].

2.9.2.3 Llamada a procedimiento remoto (XML-RPC)

Es una especificación y un conjunto de implementaciones que permite a aplicaciones de software corriendo en diferentes sistemas operativos y en diferentes ambientes hacer llamadas a procedimientos a través de Internet [52].

Es llamada a procedimientos remotos usando HTTP como transporte y XML como codificación. XML-RPC está diseñado para ser lo más sencillo posible, permitiendo al mismo tiempo complejas estructuras de datos que deben transmitirse, ser procesados y devueltos [52].

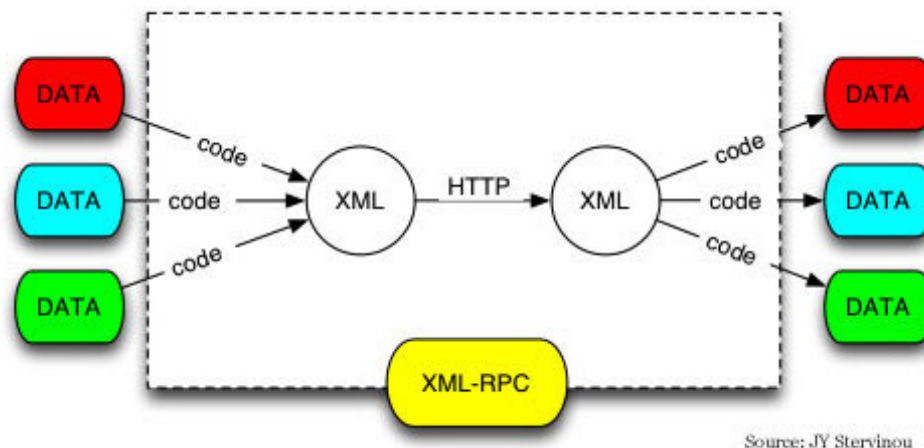


Figura 2-7 Llamada a procedimientos remotos [52]

2.10 Conclusión

Al finalizar este capítulo hemos cubierto las técnicas y herramientas de software para: solicitar información al usuario, y para la búsqueda y extracción de información. Así mismo hemos revisado las metodologías de recomendación de información y las técnicas de interoperabilidad entre aplicaciones de software.

Por último explicamos las métricas utilizadas en los sistemas de recomendación y realizamos un análisis técnico de las herramientas con las que contamos para resolver el problema.

CAPÍTULO 3

ANÁLISIS DE LA SOLUCIÓN

En este capítulo se detallan las necesidades y requerimientos de los usuarios del sistema. Se realiza también un análisis de los casos de uso del sistema. Además se efectúa un análisis del tipo de recomendación. Por último, se evalúan las alternativas y se realiza la selección de la solución.

3.1 Necesidades y requerimientos

El problema general que se busca resolver con la realización de este trabajo de tesis es la sobrecarga de información existente de música, videos y sitios web, como se muestra en el diagrama de análisis del problema a continuación.

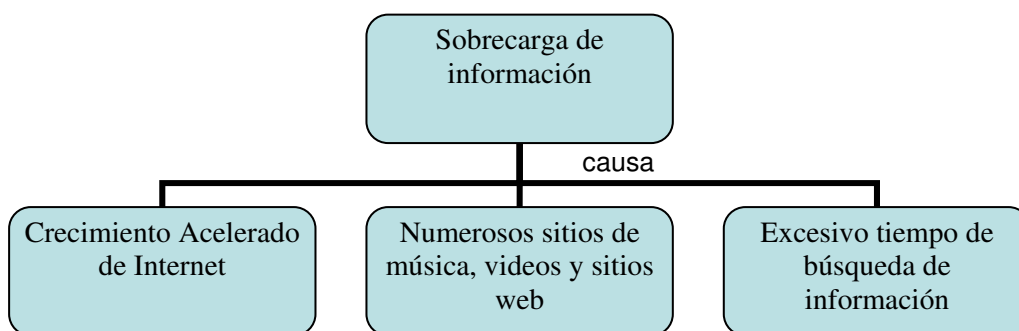


Figura 3-1 Diagrama de análisis del problema

Las necesidades de los usuarios originadas como resultado de este problema de sobrecarga de información se reflejan en los requerimientos detallados a continuación. Los requerimientos funcionales listan las tareas concretas que deben ser satisfechas por el sistema; y los no funcionales, así como las características generales del sistema.

3.1.1 Requerimientos funcionales

1. Brindar al usuario recomendaciones de música, vídeos y sitios Web, basados en su historial de navegación.
2. Brindar una interfaz de notificación de recomendaciones, integrada en el navegador Mozilla Firefox, que es el navegador Web de código libre más utilizado al momento del inicio del desarrollo de esta tesis.
3. Permitir al usuario calificar las recomendaciones según su agrado.
4. Permitir al usuario recibir recomendaciones de música, vídeos y sitios Web según las preferencias que haya tenido al calificar recomendaciones previas.

3.1.2 Requerimientos no funcionales

1. Evitar la interrupción de las actividades de navegación del usuario.
2. El sistema se integrará al navegador Web Mozilla Firefox.

3.2 Análisis de casos de uso

El modelo de casos de uso muestra información no trivial sobre el comportamiento dinámico y la secuencia de transacciones del sistema.

Debido a que para la implementación del servidor del sistema se utilizará un lenguaje de programación orientado a objetos, para el análisis y el diseño del sistema se utilizará el lenguaje de modelamiento unificado (UML), que es un lenguaje para especificar, visualizar, construir y documentar los sistemas desarrollados con lenguajes orientados a objetos.

Los casos de uso describen los requerimientos funcionales del sistema a alto nivel, desde la perspectiva de los actores típicos. Cada caso de uso describe un comportamiento del sistema observable, el mismo que es una transacción de valor para el actor.

Los actores son los usuarios y pueden ser personas, máquinas u otros sistemas. Los actores participan en los casos de uso cuando realizan un trabajo significativo para el sistema.

3.2.1 Diagrama general de casos de uso

El sistema se divide en dos partes: la aplicación cliente que se ejecuta en el navegador Web del computador del usuario, y, el servicio Web que proporcionará la funcionalidad necesaria cuando el cliente lo requiera.

Al realizar el análisis de la solución hemos obtenido tres casos de uso generales, los cuales son los principales considerados en el sistema:

- Obtener recomendación
- Enviar calificación
- Recibir retroalimentación

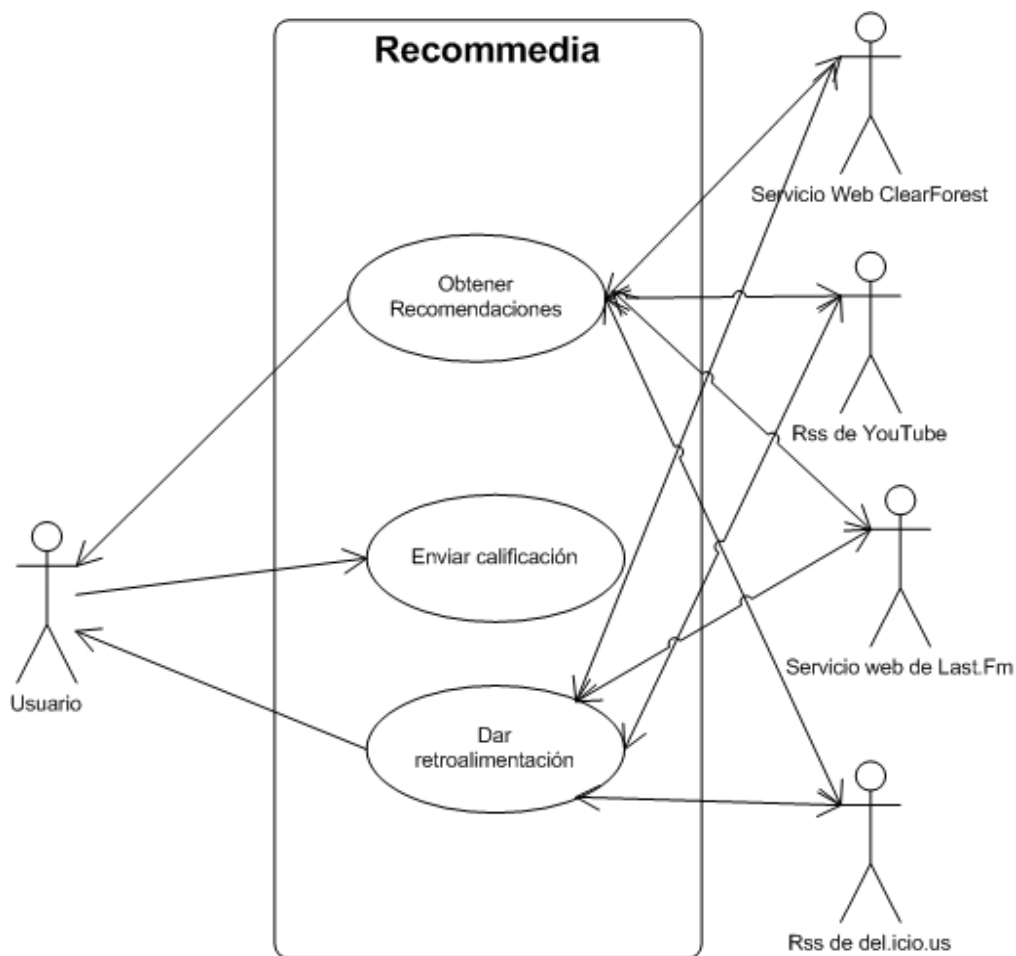


Figura 3-2 Diagrama de casos de uso

Para cada uno de estos contextos se describe la funcionalidad requerida en el sistema necesaria para cumplir con los objetivos propuestos en esta tesis.

3.2.2 Descripción de casos de uso

A continuación se detallarán las especificaciones de cada uno de los casos de uso de la aplicación.

Caso de uso 1: Obtener recomendaciones

El usuario recibirá recomendaciones de tres tipos: música, vídeos o sitios Web, según su historial de navegación utilizando el navegador Web.

El historial de navegación del usuario servirá para obtener las entidades básicas extraídas de las páginas Web que visita, utilizando el Servicio Web de ClearForest, con lo cual el sistema solicitará información a mashups o servicios Web para obtener sugerencias de música, vídeos y enlaces, respectivamente; luego, el sistema mostrará esta información al usuario por medio de una extensión en el navegador.

Hemos decidido utilizar ClearForest ya que el Servicio Web que proporciona de manera gratuita provee la posibilidad de obtener entidades a partir de páginas web, que es lo que necesitamos para poder generar el proceso de recomendación.

Escenarios:

1.1 El usuario obtiene recomendación de un tipo específico

El usuario puede decidir escoger entre los tres tipos de recomendaciones ofrecidas por el sistema: música, videos y sitios Web. Cuando el usuario ha escogido una de estas opciones, la extensión le proporcionará avisos de recomendaciones específicas para la opción seleccionada, dependiendo de la información obtenida en su historial de navegación.

1.2 El usuario obtiene recomendaciones basadas en calificaciones previas

Cuando el sistema de recomendación no ha obtenido resultados basados en las entidades encontradas en el historial de navegación, procede a utilizar el historial de las calificaciones proporcionadas por el usuario para generar recomendaciones.

1.3 El usuario no obtiene recomendaciones

Como se ha visto, el sistema tiene dos formas de generar recomendaciones: la primera es utilizando el historial de navegación del usuario, y la segunda, basándose en calificaciones previas. Cuando el sistema ha agotado estas dos opciones y el usuario no ha tenido actividad de navegación, ya no se generarán recomendaciones.

Caso de uso 2: Enviar calificación

El usuario podrá ponderar o calificar las recomendaciones recibidas en un rango que indique su interés o desinterés por una u otra recomendación recibida.

La calificación se realizará con el objetivo de conocer los gustos del usuario, a través de la selección de la recomendación del usuario por medio de diferentes acciones, como, por ejemplo, dar un clic al enlace que mostrará la recomendación en el navegador, borrar la recomendación o calificarla explícitamente de manera positiva o negativa.

Escenarios:

2.1 Enviar calificación positiva

El usuario podrá calificar las recomendaciones que considere de su interés. Hay dos alternativas que podrán ser consideradas como calificaciones positivas: Cuando el usuario abre un enlace de una recomendación generada por el sistema, y, cuando el usuario realiza una calificación positiva explícita. Estas calificaciones servirán para que el sistema genere nuevas recomendaciones.

2.2 Enviar calificación negativa

El usuario también podrá enviar calificaciones negativas. Serán consideradas negativas aquellas recomendaciones que el usuario haya borrado y las que haya calificado explícitamente como negativas.

2.3 No enviar calificación

El usuario puede decidir no enviar calificaciones explícitas, en las que expresa su gusto o disgusto por determinado ítem. Sin embargo, esto no evita que se envíen calificaciones implícitas encapsuladas en las acciones de abrir un enlace de recomendación y borrar dicha recomendación.

Caso de uso 3: Dar retroalimentación

La retroalimentación permitirá al sistema construir un perfil del usuario basado en sus intereses. Una vez que el usuario ha calificado las recomendaciones recibidas, el sistema ha recibido la retroalimentación necesaria para generar nuevas recomendaciones, basándose en los gustos del usuario.

Escenarios:

3.1 El sistema genera recomendaciones basado en retroalimentación

Cuando el usuario ha calificado positivamente las recomendaciones, el sistema tiene la información necesaria para generar nuevas recomendaciones basadas en los gustos del usuario.

3.2 El sistema no genera recomendaciones basado en retroalimentación.

Este escenario se puede dar por dos razones. La primera, cuando el usuario no ha calificado positivamente las recomendaciones, y la segunda cuando el sistema no ha podido generar recomendaciones basadas en las entidades encontradas en las recomendaciones que han sido del gusto del usuario.

3.3 Análisis del tipo de recomendación

Los sistemas de recomendación ayudan al usuario a escoger elementos de una gran cantidad de opciones. El volumen de la información es la razón principal del surgimiento de estos sistemas.

Un reto que se presenta indudablemente al momento de crear un sistema de recomendación, es el de definir el tipo de recomendación.

La recomendación de los ítems puede hacerse basándose en diversos métodos.

La recomendación por similitud de objetos se caracteriza porque el perfil creado para representar al usuario se basa en el análisis del contenido de los objetos, con el fin de recomendar objetos similares a los usuarios.

La recomendación social es aquella que busca características sociales similares entre los usuarios para recomendar ítems. Hay diversos tipos de recomendaciones que se pueden realizar, según el perfil de los usuarios o según las características de los ítems que son de su preferencia.

En la recomendación basada en historia, algunos sistemas mantienen una lista de compras, el historial de navegación en la World Wide Web o el contexto de correos electrónicos como un perfil de usuario. Adicionalmente, también es común mantener la retroalimentación relevante del usuario asociado con cada ítem en el historial.

Hemos decidido utilizar la recomendación basada en historia puesto que en el contexto en el que es utilizado el sistema, es recomendable

almacenar el historial de navegación del usuario con el fin de obtener sus preferencias e intereses y con ellos obtener la información necesaria para generar recomendaciones.

3.4 Análisis de las alternativas y selección de la solución

De acuerdo al análisis técnico realizado anteriormente y a las herramientas disponibles para implementar el sistema de recomendación y considerando también que la motivación principal para usar puntuaciones implícitas es que remueve el costo al evaluador de examinar y asignar puntajes a los ítems, se requiere de una herramienta que permita obtener las direcciones URL que el usuario visita mediante un navegador de código abierto, como lo es Mozilla Firefox. Slogger se ajusta perfectamente a las características necesarias, pues permite almacenar el historial de navegación del usuario y su uso es mediante una extensión para Mozilla Firefox.

Se requiere además de una herramienta que permita obtener las entidades dada una página Web, esta funcionalidad la tiene ClearForest mediante su servicio Web que es de acceso libre, llamado Semantic Web Services.

Para poder obtener información basada en las entidades que permita generar recomendaciones a los usuarios requerimos de la utilización de mashups o servicios Web que proporcionen información de música, vídeos y enlaces. En la Web encontramos tres sitios ampliamente conocidos por facilitar a los usuarios interactuar en estos tópicos, estos sitios Web son Last.fm, YouTube, y del.icio.us, los cuales permiten a los desarrolladores acceder a parte de sus funcionalidades a través de servicios Web. Si bien hay otros sitios que proporcionan este tipo de información, Last.fm, YouTube y del.icio.us poseen robustas interfaces de programación de la aplicación (API) que permitirán que nuestro sistema interactúe fácilmente con ellos. Inclusive, Last.fm provee una librería de código abierto creada en .Net que sirve como un puerto para acceder a toda la funcionalidad provista por los servicios Web de Last.fm. Además, YouTube y del.icio.us cuentan con el servicio de RSS para obtener videos o enlaces relacionados con categorías de interés de los usuarios.

Además de las aplicaciones descritas anteriormente, procederemos a analizar las herramientas que serán utilizadas para el desarrollo del sistema.

A continuación se muestra un cuadro comparativo de los lenguajes de programación más importantes para desarrollar aplicaciones.

	Visual Basic	Visual Basic .Net	Visual C# .Net	Java
Requerimientos	Ninguno	.Net Framework 2.0	.Net Framework 2.0	Java Virtual Machine
Nivel de Conocimiento	Básico	Avanzado	Intermedio	Básico
Orientado a Objetos	No	Si	Si	Si

Tabla 3-1 Cuadro comparativo de los lenguajes de programación más importantes para desarrollar aplicaciones.

Como nuestra aplicación va a estar compuesta también por el servicio Web que va a recibir solicitudes de los clientes y enviar información a otros servicios Web, al analizar los lenguajes de programación en los que dicho servicio Web podría ser desarrollado, consideramos la mejor alternativa utilizar Visual Basic .NET, debido al nivel de conocimientos que se tiene, por lo que se necesita una herramienta de desarrollo que soporte .NET Framework 2.0.

Entre las herramientas que cumplen los requerimientos están: Microsoft Visual Studio .Net 2005 y Microsoft Visual Studio .NET 2008. De entre las herramientas nombradas se utilizará Visual Studio .NET 2005 debido a que es la versión disponible al momento del inicio de esta tesis.

3.5 Conclusión

En este capítulo hemos realizado el análisis de la solución para el problema presentado en esta tesis.

Incluimos un análisis de los requerimientos funcionales y no funcionales provenientes de las necesidades de los usuarios. Se realizó el análisis de casos de uso del sistema, detallando el diagrama respectivo y haciendo una descripción de cada caso.

Además analizamos tres diferentes tipos de recomendación, por similitud de objetos, social y basada en historial. Y explicamos los motivos por los cuales seleccionamos la recomendación basada en historial.

Finalmente revisamos las posibles opciones para la solución del problema, tanto en el mecanismo a emplear como en las herramientas a utilizar en el mismo.

CAPÍTULO 4

DISEÑO

El diseño del sistema de recomendación parte de los lineamientos dados por el análisis, descrito en el capítulo anterior, y expone un bosquejo general de la solución, sin profundizar en los detalles de su implementación.

En este capítulo se describe el diseño general, a partir del cual se realizan el diseño de la recomendación, la base de datos, el diagrama de clases y por último los diagramas de secuencias relevantes.

4.1 Diseño general

Esta sección abarca los planteamientos generales de la lógica utilizada para generar recomendaciones y recibir retroalimentación del usuario.

A partir de estos planteamientos realizamos el diseño de la recomendación, de la base de datos, el diseño de los módulos del sistema, el diseño de los diagramas de secuencia y el diseño de los diagramas de clases.

En la figura 4-1, que se muestra a continuación, está el diagrama que esquematiza la arquitectura del sistema.

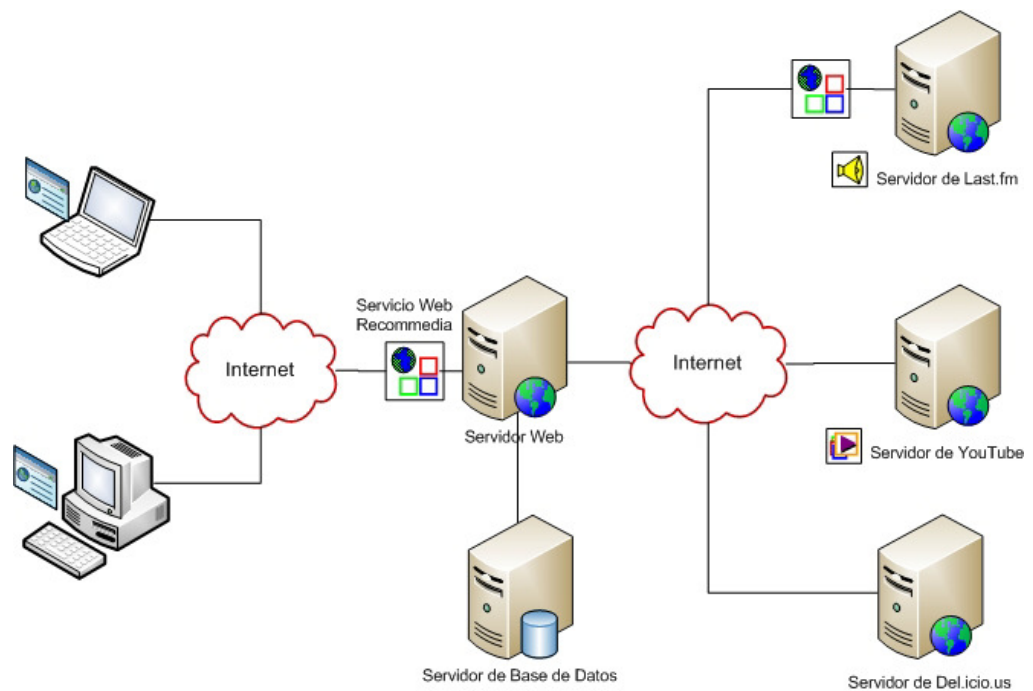


Figura 4-1 Arquitectura del sistema

De manera general, el sistema de recomendación se puede dividir en dos partes principales:

- Aplicación cliente en Mozilla Firefox.
- Servicio Web de recomendaciones.

La parte correspondiente al cliente utilizará una arquitectura de tres módulos:

Módulo de presentación: Representa la interacción con el cliente que usa un navegador Web. Contendrá las características gráficas en Mozilla Firefox que permitirán al usuario interactuar con la aplicación.

Módulo de lógica: Encargado de realizar los procedimientos necesarios para las operaciones que el sistema soporta.

Módulo de datos: Representa los datos adquiridos de la interacción entre el sistema y los usuarios. El objetivo de esta capa es servir de repositorio y brindar acceso a información necesaria en el momento requerido.

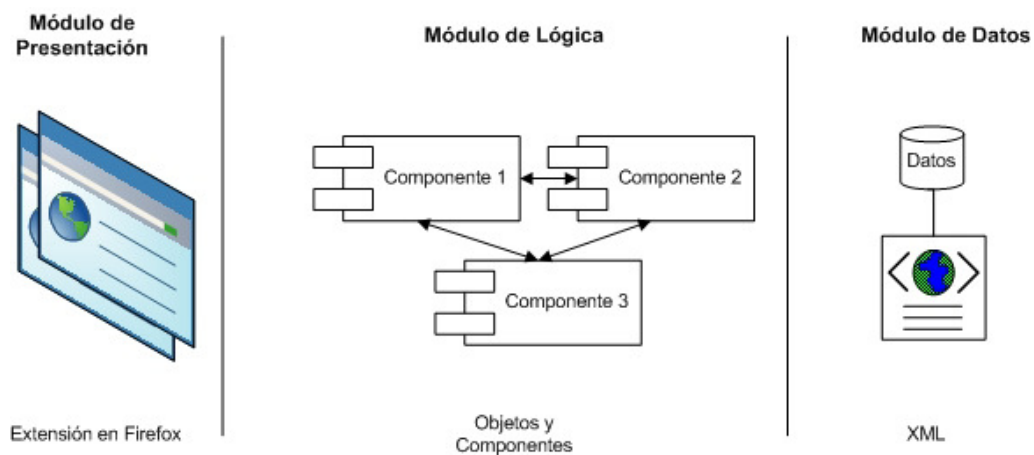


Figura 4-2 Módulos del cliente

Para construir la parte correspondiente al servidor, hemos decidido utilizar una arquitectura orientada a servicios, con el fin de que exista flexibilidad para adaptar cambios futuros e interoperabilidad con los componentes del cliente. También se utilizara el estilo de programación por capas, con el objetivo de separar la lógica de negocios de la lógica de diseño y datos. Se han considerado las siguientes capas en la aplicación servidor:

Capa de servicio: Encargado de exponer los métodos Web para ser consumidos por la extensión.

Capa de lógica de negocio: Es donde se establecen todas las reglas del negocio que se deben cumplir.

Capa de acceso a datos: Esta capa es la encargada de acceder a los datos. Está formada por gestores de bases de datos que realizan el almacenamiento de los datos.

4.2 Diseño de la recomendación

Cada operación de recomendación realizada por el sistema se encarga de obtener las entidades relevantes provenientes de la información de

las páginas Web para luego generar las recomendaciones según la información obtenida de diferentes servicios especializados en áreas específicas como son YouTube, Last.fm y Del.icio.us.

El proceso de retroalimentación empieza cuando el usuario pondera o califica las recomendaciones recibidas en el navegador Mozilla Firefox. Las calificaciones pueden ser positivas, negativas o neutras.

Por defecto, las recomendaciones nuevas y sin ponderar tienen un valor neutro, identificado por el número 0. Asumiendo que las calificaciones positivas son aquellas que han sido del agrado del usuario y las calificaciones negativas aquellas que no han satisfecho las necesidades o expectativas del usuario, hemos establecido un rango de calificación basado en puntuaciones.

Puntos	Acción	Grado
2	Calificación explícita (positiva)	MEJOR
1	Abrir enlace	BUENO
0	Ninguna	NEUTRO
-1	Borrar recomendación	MALO
-2	Calificación explícita (negativa)	PEOR

Tabla 4-1 Rangos positivos y negativos

Esta ponderación se ha realizado basándose en la intuición y en el análisis de las acciones del usuario apoyadas en sus intereses. Para las calificaciones positivas, la acción de abrir un enlace es considerada con menor valor que la acción que el usuario realiza al establecer una recomendación como buena puesto que cuando el usuario abre un enlace muestra su interés por conocer el contenido, pero este contenido no necesariamente será de su agrado. Al calificar explícitamente la recomendación como buena, el usuario muestra un interés genuino basado en sus gustos.

En el caso de las calificaciones negativas, la acción de borrar una recomendación tiene una menor puntuación que aquella acción realizada explícitamente por el usuario al calificar una recomendación como mala, esta afirmación es basada en fundamentos similares a los expuestos en las calificaciones positivas.

4.3 Diagrama de clases

En la figura a continuación se encuentra el diagrama de clases de la parte correspondiente al servidor de Recommedia. Este diagrama estático describe la estructura de Recommedia mostrando sus clases y las relaciones entre ellas.

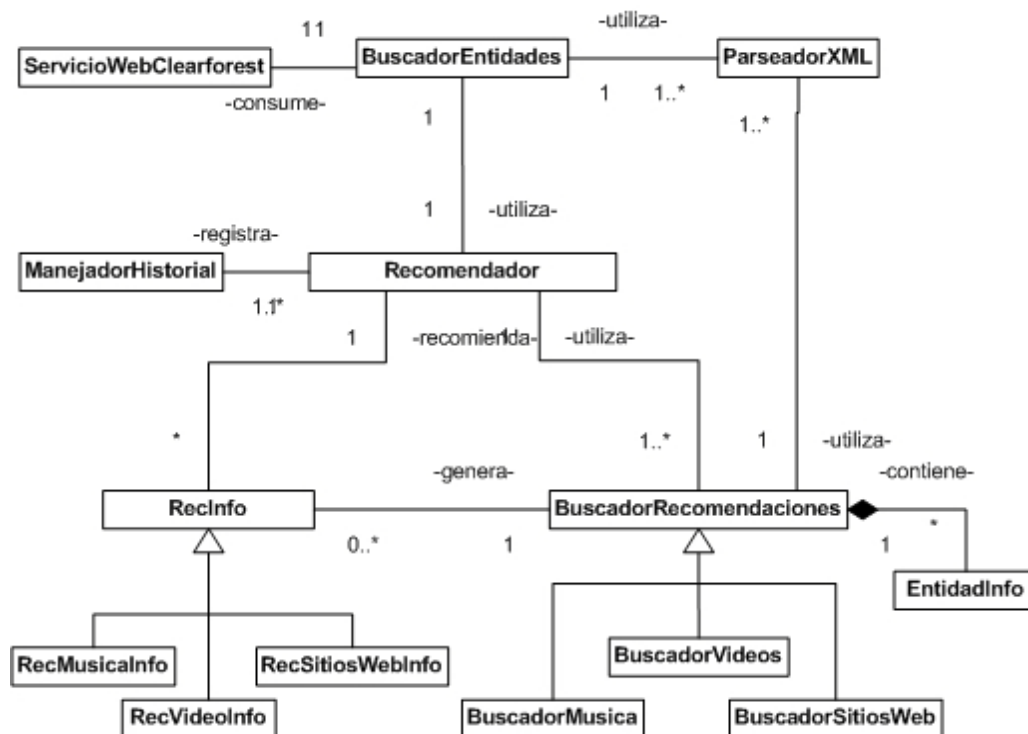


Figura 4-3 Diagrama de clases de Reommedia

En el proceso de recomendación, la clase Recomendador se encarga de gestionar las recomendaciones y la retroalimentación, pues expone los métodos que contendrá el servicio Web.

Cuando una petición de recomendación es realizada por el cliente, Recomendador utiliza a BuscadorEntidades para enviar la información de la página HTML que el usuario ha navegado, según el historial almacenado en el cliente, al Servicio Web de ClearForest. Dicho servicio se encarga de devolver las entidades que ha encontrado utilizando semántica Web.

Cuando nuestro servicio obtiene las entidades, las analiza con el fin de obtener la información de interés para la aplicación, con la cual, a través de `BuscadorRecomendaciones`, realiza peticiones a los servicios Web de Youtube, Del.icio.us y Last.fm. Todos estos servicios devuelven resultados con información de vídeos, enlaces o canciones. Nuevamente esta información es analizada, utilizando la clase `ParseadorXML`, y encapsulada en un objeto `RecInfo` que puede ser de tres tipos: `RecMusicalInfo`, `RecVideoInfo` o `RecSitiosWebInfo`, según el tipo de información devuelta. Finalmente la extensión obtiene la lista de recomendaciones y muestra la información.

Al realizar el proceso de calificación, el cliente, a través de la extensión, envía información relevante sobre sus gustos, según las acciones que realice, como se describe en la arquitectura general. Esta información es utilizada para filtrar las recomendaciones enviadas a los usuarios y generar nuevas recomendaciones a partir de aquellas con mayor valor para el usuario.

4.4 Diagramas de interacción de objetos

Los diagramas de interacción representan la forma en que los objetos se comunican entre sí al realizar peticiones a un evento. Utilizamos los diagramas de secuencia, un tipo de diagrama de interacción, para mostrar las interacciones entre los objetos.

A continuación, se muestran los diagramas de secuencia de Recommedia, agrupados según el caso de uso y según el ámbito: Cliente y Servidor.

Caso de Uso: 1.- Obtener recomendaciones

Cliente

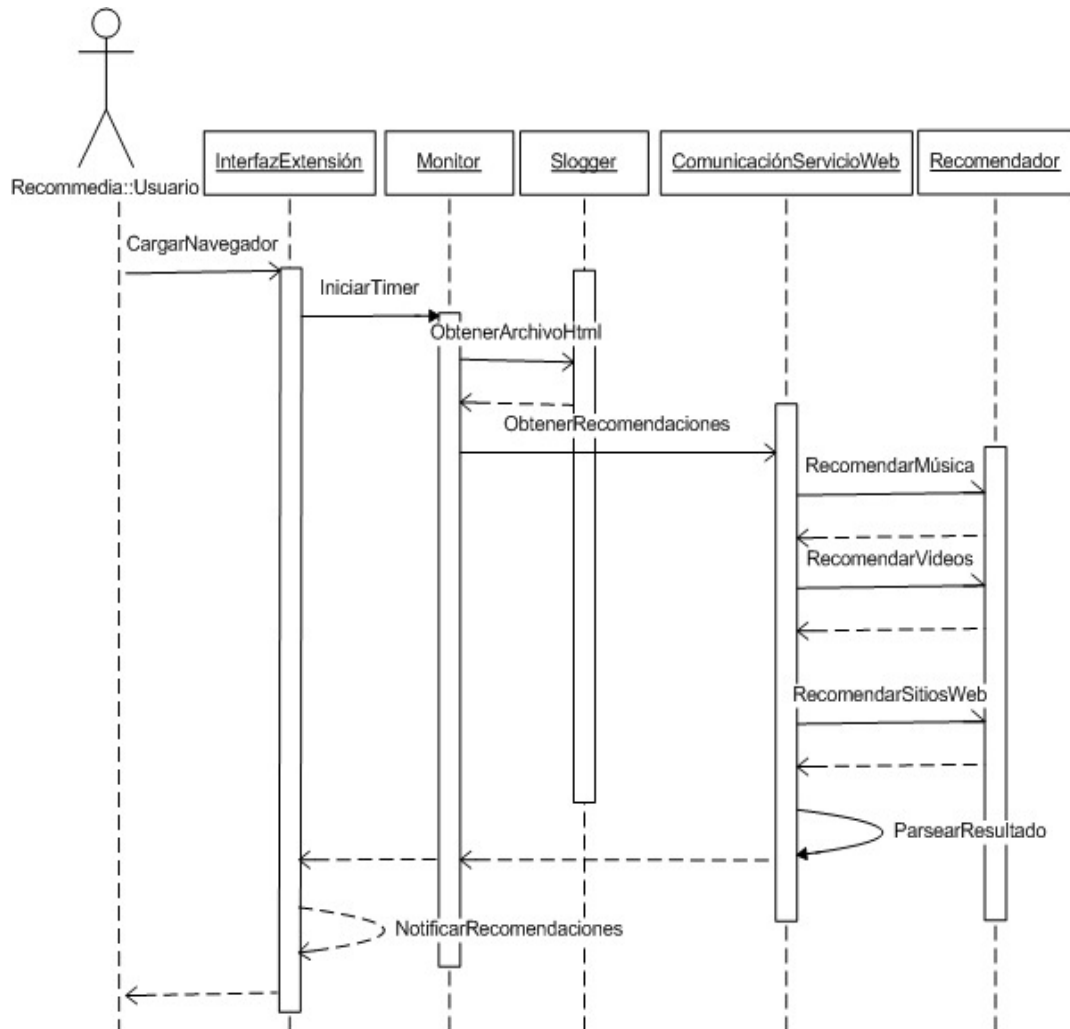


Figura 4-4 Diagrama de secuencia: obtener recomendaciones cliente

Cada vez que el usuario utiliza el navegador Web Mozilla Firefox, se inicia un cronómetro, el cual monitorea cada intervalo de tiempo los archivos HTML que la extensión de Slogger ha guardado, correspondientes a las páginas navegadas por el usuario. El Monitor utiliza a

ComunicaciónServicioWeb para enviar estos archivos HTML a Recomendador, el servicio Web de Recommedia.

Cuando el servicio ha respondido, la clase de comunicación se encarga de parsear el resultado obtenido, para ser almacenado en un archivo XML que contiene las recomendaciones del día. Luego, el Monitor se encarga de mostrar las recomendaciones recibidas a través de la barra lateral, y también de notificar de la existencia de nuevas recomendaciones.

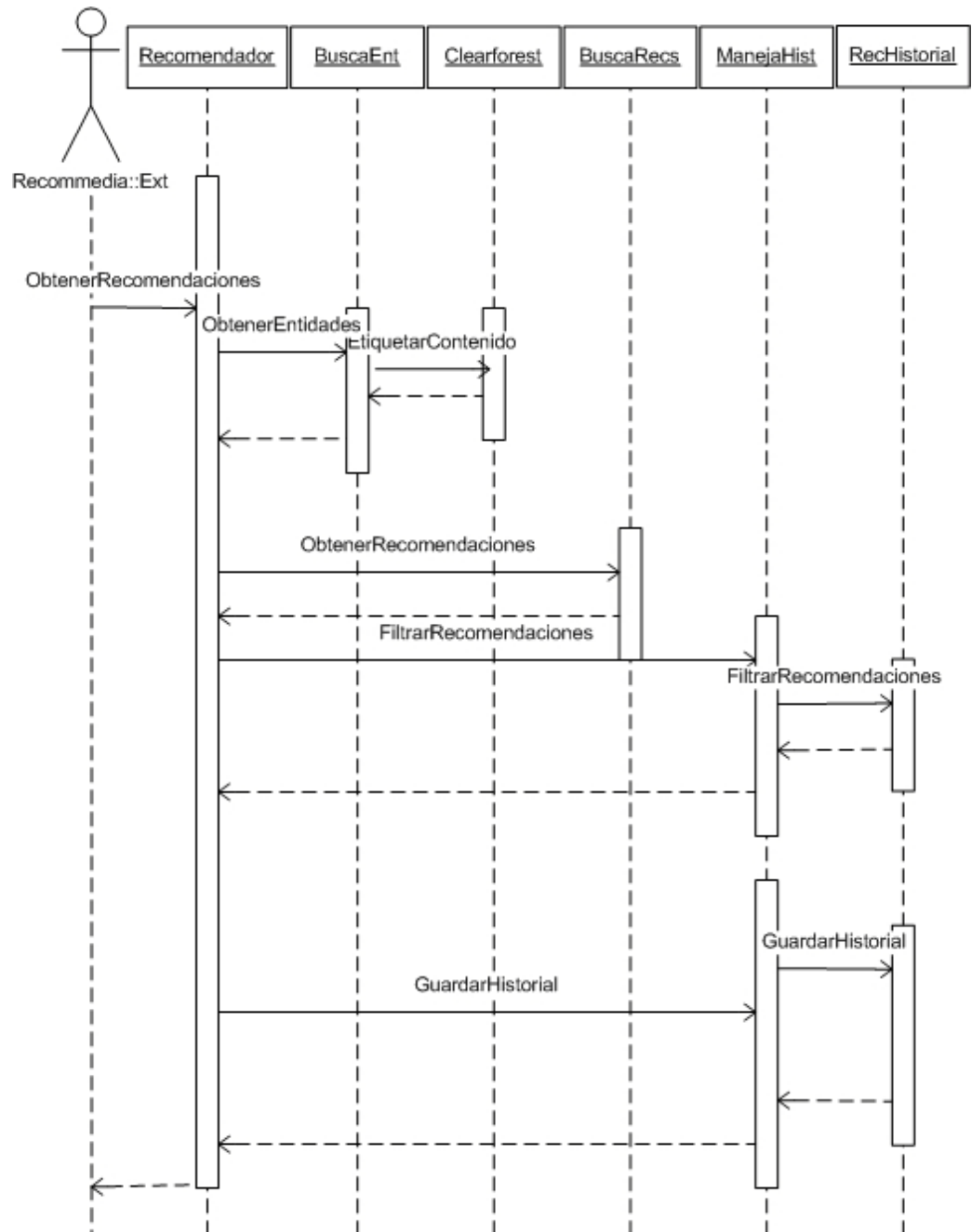
Servidor

Figura 4-5 Diagrama de secuencia: obtener recomendaciones servidor

Cuando la extensión en Firefox ha realizado una petición de recomendaciones al servicio Web, se realizan las siguientes acciones:

1. – Obtener entidades: Se hace una llamada a BuscadorEntidades, el cual llama al servicio Web de ClearForest, este expone un método que recibe el texto, en nuestro caso en formato HTML, y cuyos parámetros de salida son entidades. A través de la clase ParseadorXML se procede a parsear las entidades con el fin de obtener la información necesaria para nuestros fines.
2. – Obtener recomendaciones: Dadas las entidades, se utiliza la clase BuscadorRecomendaciones para hacer llamadas a los servicios Web que proveen información de música, vídeos y sitios Web.
3. – Filtrar Recomendaciones. Se utilizan las clases ManejadorHistorial y RecHistorial para filtrar las recomendaciones y evitar que el usuario reciba recomendaciones repetidas.
4. – Guardar Historial. Las nuevas recomendaciones son guardadas en la base de datos y finalmente son enviadas al cliente que realizó la petición Web.

Caso de Uso: 2.- Enviar calificación

Cliente

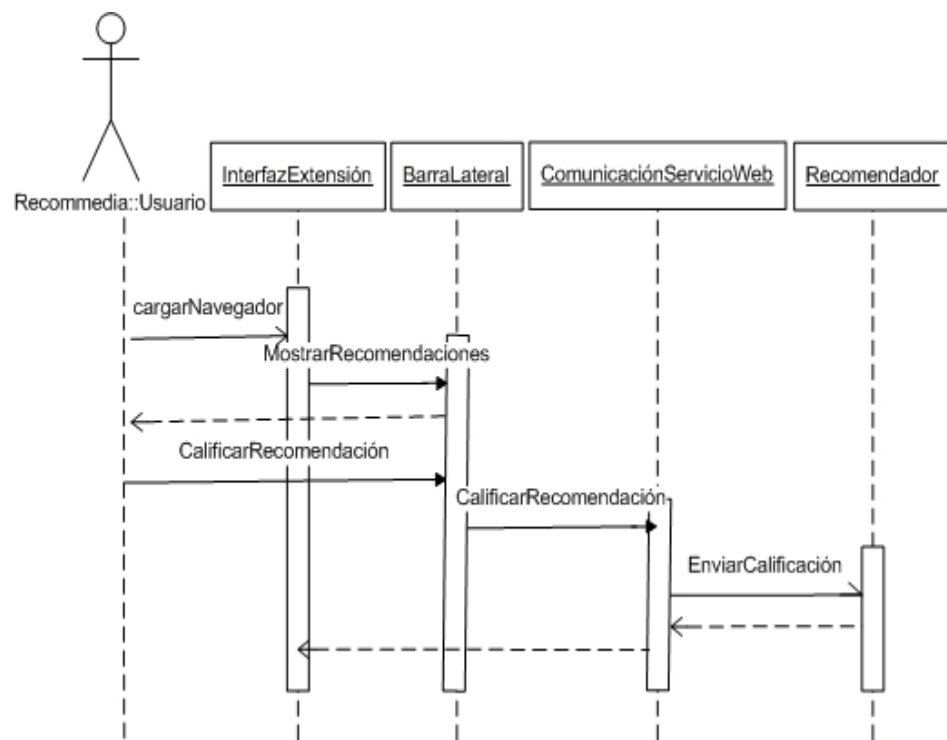


Figura 4-6 Diagrama de secuencia: enviar calificación cliente

Cuando el usuario decide mostrar las recomendaciones que se visualizan en la barra lateral, puede calificarlas al hacer click en diferentes opciones, se utiliza la clase ComunicaciónServicioWeb para hacer la llamada al servicio Web Recomendador.

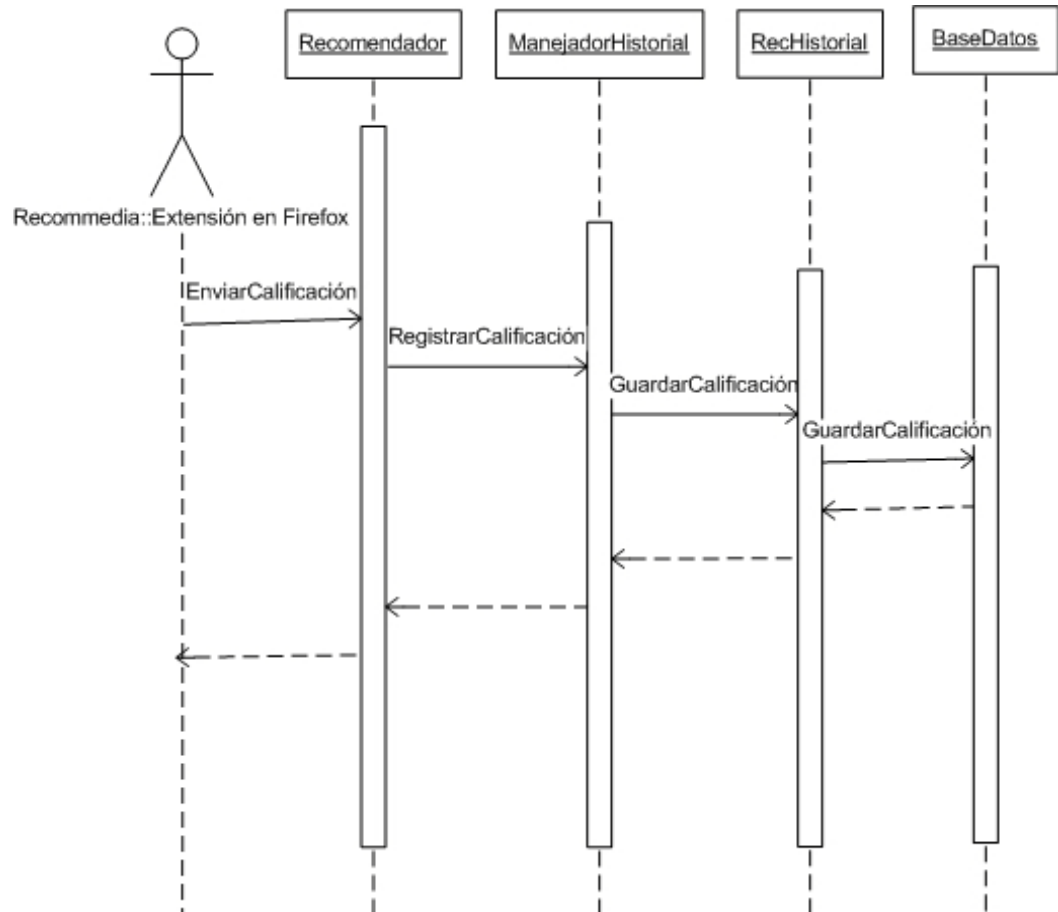
Servidor

Figura 4-7 Diagrama de secuencia: enviar calificación servidor

Cuando el usuario envía una calificación a través de la extensión en Firefox, realizando cualquiera de las acciones especificadas en la tabla 4-1, la clase Recomendador llama a ManejadorHistorial, este llama a la clase de acceso a datos RecHistorial, que se encarga del

almacenamiento en la base de datos de Recommedia para luego servir de filtro en el envío de recomendaciones.

Caso de uso 3: Dar retroalimentación

Cliente

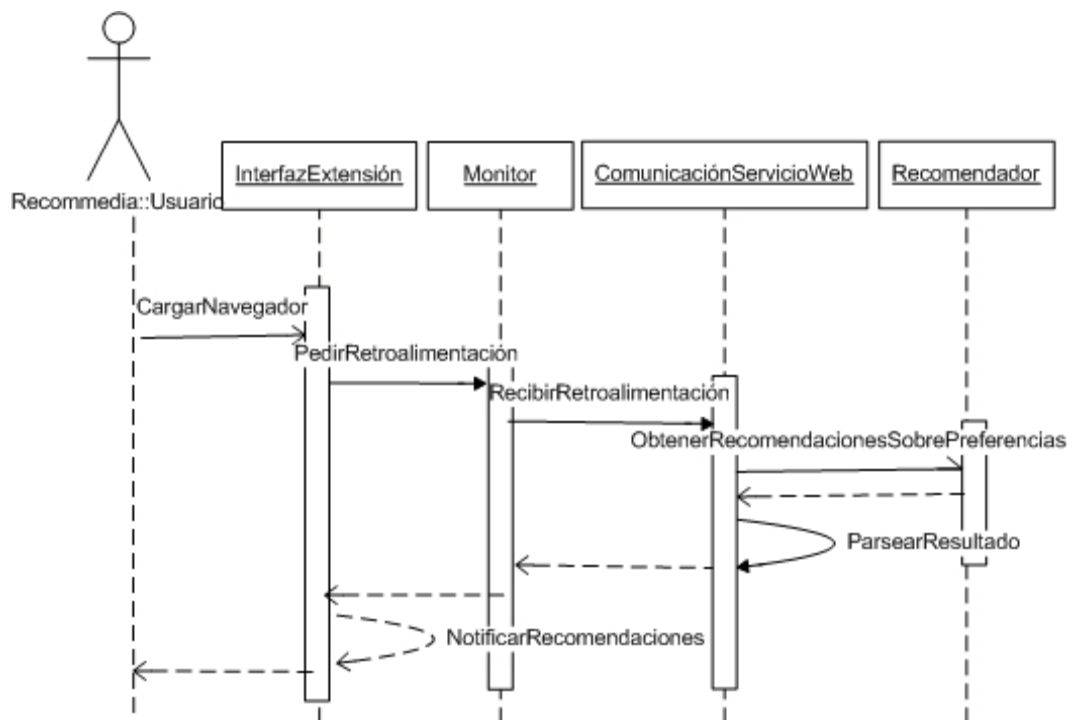


Figura 4-8 Diagrama de secuencia: dar retroalimentación cliente

Los resultados de la retroalimentación recibida por Recommedia son solicitados por el usuario cuando no hay recomendaciones nuevas que realizar. La clase Monitor hace la petición a la clase

ComunicaciónServicioWeb, la cual se encarga de realizar la llamada al método Web de Recomendador.

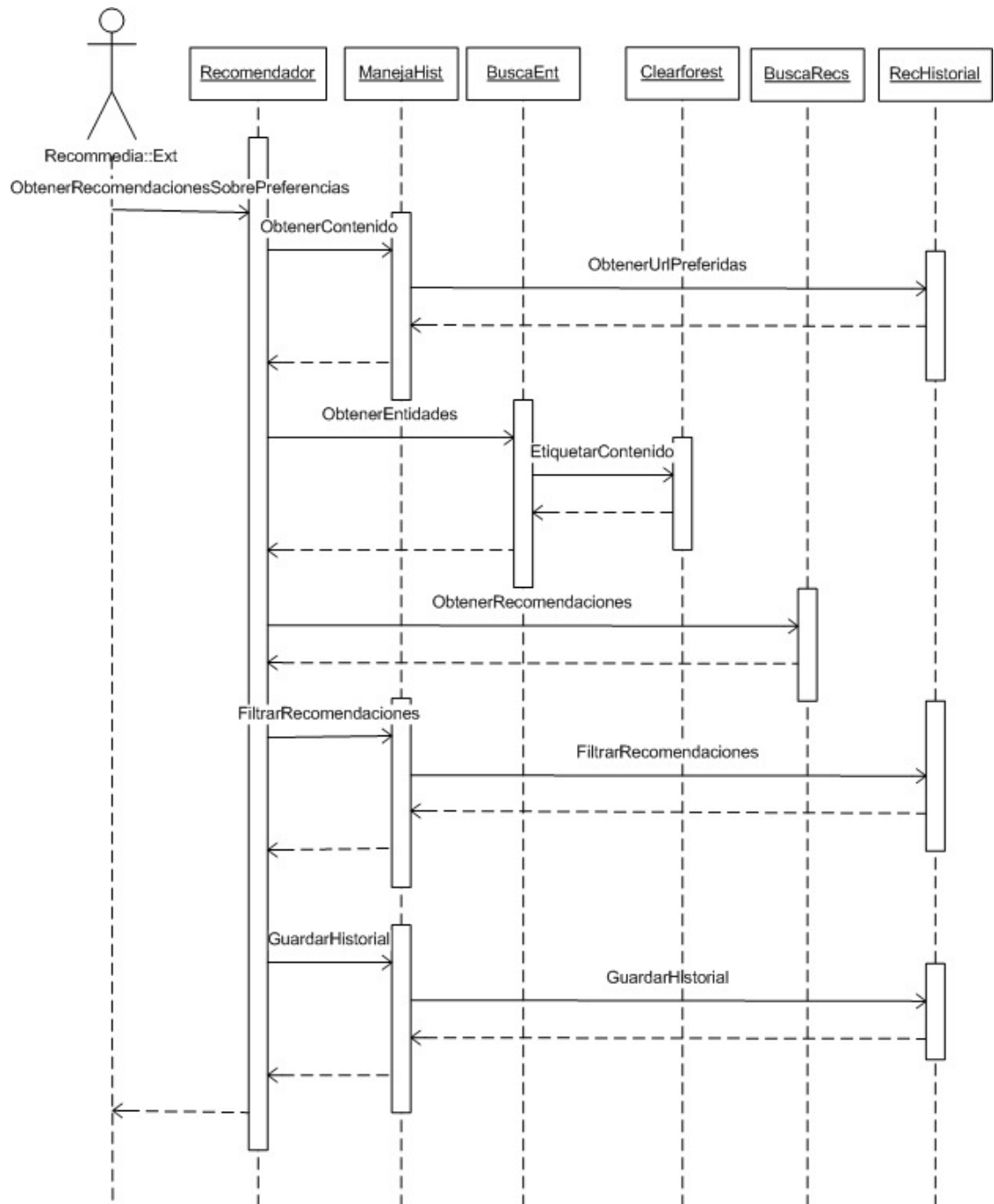
Servidor

Figura 4-9 Diagrama de secuencia: dar retroalimentación servidor

Desde el punto de vista del servidor, al recibir una petición del cliente, se procesa la retroalimentación. A través de la clase `ManejadorHistorial` se obtienen las url que han tenido una mayor ponderación, utilizando la clase `RecHistorial` para acceder a la base de datos. Luego, la clase `BuscadorEntidades` se encarga de hacer la llamada al servicio Web de `ClearForest` para obtener las entidades, que luego van a ser parseadas para extraer la información relevante para nuestro sistema. Una vez obtenidas las entidades, se procede a utilizar la clase `BuscadorRecomendaciones` para recibir las recomendaciones de los servicios Web de `YouTube`, `Del.icio.us` y `Last.fm`. Las recomendaciones son filtradas utilizando la clase `ManejadorHistorial`, que requerirá a `RecHistorial` para acceder a la base de datos y verificar que las recomendaciones no sean repetidas. Una vez realizada la verificación, se procede a guardar el historial de recomendaciones recibidas en la base de datos y posteriormente, enviarlas al cliente.

4.5 Diseño de la base de datos

Nuestro sistema requiere de una base de datos alojada en el proveedor del servicio, la cual permitirá almacenar la información relacionada a las recomendaciones. Esta información proviene principalmente de dos fuentes:

- Las recomendaciones generadas por el servicio Web.
- La retroalimentación proporcionada por el usuario al interactuar con el sistema.

El diseño de la base de datos es sencillo, teniendo en cuenta que gran parte de la complejidad en la realización del sistema se concentra en el servicio Web y en la extensión para Mozilla Firefox.

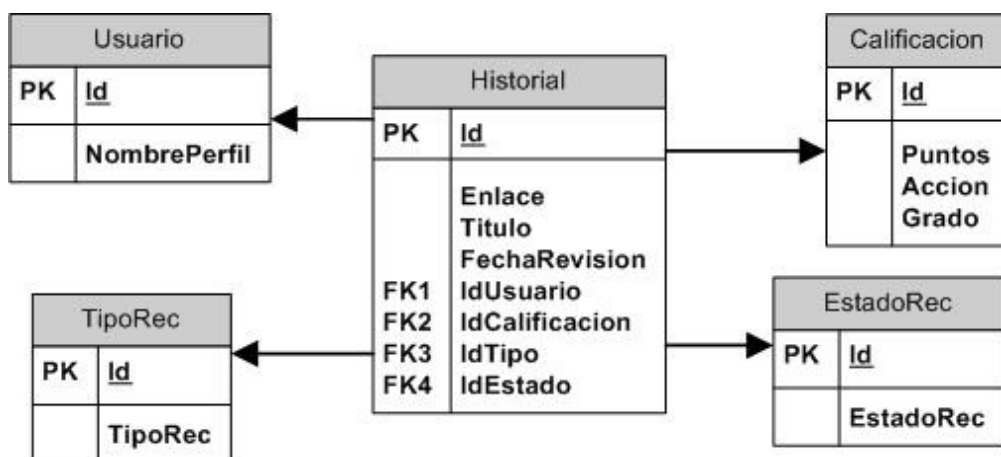


Figura 4-10 Diseño base de datos Recomendía

La ilustración anterior muestra el diseño de la base de datos para nuestro sistema, la información que allí se almacenará corresponde a la preferencia del usuario al momento de interactuar con las recomendaciones a él presentadas, tal como lo explicamos en la primera parte de este capítulo.

La base de datos de Recommedia constará de cinco tablas:

Historial: Es la tabla principal, en donde se almacenarán las recomendaciones realizadas, junto con la calificación asignada al usuario. Dichas recomendaciones pasarán a estado inactivo una vez que el usuario las haya calificado.

Usuario: Esta tabla almacena las identificaciones de los usuarios del sistema de recomendación. Cada usuario estará identificado por el perfil que Mozilla Firefox crea para él.

Tipo de Recomendación: Esta tabla alberga los tres tipos de recomendaciones que nuestro proyecto de tesis comprende: Música, vídeos y sitios Web.

Estado de la Recomendación: Básicamente, se han considerado dos estados: activo e inactivo.

Calificación: En esta tabla se encontrarán las descripciones y ponderación del rango de calificaciones empleado para cuantificar el interés del usuario en las recomendaciones recibidas.

4.6 Conclusión

Hemos revisado el diseño del sistema de recomendación el cual comprende el diseño general del sistema, del cual se derivan el diseño de la recomendación, el diseño de la base de datos, el diagrama de clases y los diagramas de secuencia más importantes. Todos en conjunto permiten comprender la estructura y funcionamiento de nuestro sistema.

Realizar el diseño de un sistema de recomendación implica la utilización de métodos para definir los gustos e intereses del usuario, asegurar la persistencia de la información y también realizar la coordinación y cooperación de cada una de las partes del sistema.

CAPÍTULO 5

IMPLEMENTACIÓN

En este capítulo se hace una descripción del proceso de implementación de la aplicación demostrativa de este proyecto de tesis una vez realizadas las etapas de análisis y diseño del sistema.

Se describen los estándares de implementación del sistema. Además se especifican las tecnologías utilizadas y los pasos seguidos en el proceso de implementación del sistema. Luego se detallan los requerimientos de hardware y software necesarios. Finalmente se enumeran los problemas encontrados durante la implementación del sistema.

5.1 Estándares de implementación del sistema

Se han definido ciertas reglas en la programación del sistema con el fin de facilitar su mantenimiento y entendimiento. Estas reglas han sido consideradas según las guías de diseño de Microsoft, las cuales fomentan la consistencia y fácil predicción del código desarrollado en .Net y Microsoft recomienda su uso con el fin de que los desarrolladores mejoren el modelo de programación que utilizan así como también su productividad [53].

Las reglas adoptadas son:

- Utilización de sustantivos para nombrar una clase, empleando la notación Pascal, manteniendo la primera letra en el identificador y la primera letra de cada palabra subsecuente concatenada en mayúsculas.
- Para nombrar los métodos del sistema se utilizará verbos o frases verbales junto con la notación Pascal.
- Las propiedades serán nombradas utilizando sustantivos y la notación Pascal. Se evitará utilizar la notación Húngara.

- Los parámetros serán nombrados utilizando la notación Camel, procurando utilizar nombres que describan su significado en lugar de su tipo de dato.

5.2 Tecnologías

Para desarrollar la extensión en Mozilla Firefox se ha utilizado el lenguaje de programación XUL, encargado de manejar la interfaz en el navegador Web, junto con Javascript para poder implementar la funcionalidad de la extensión.

Además se utilizó XPCOM, tecnología propia de la plataforma de Mozilla para poder interactuar con servicios Web y también con el directorio de archivos de Windows. Luego, para poder vincular a XPCOM con Javascript se utilizó XPConnect, que permite la interacción entre ambos y facilita la creación de componentes en Javascript.

El servicio Web fue desarrollado utilizando el ambiente de desarrollo proporcionado por Visual Studio 2005, con el lenguaje de programación Visual Basic .Net. Para mantener la persistencia de los datos se utilizó el motor de base de datos Microsoft SQL Server 2005.

En vista de que Visual Basic .Net es un lenguaje orientado a objetos, y como se decidió en el capítulo de diseño, se procedió a implementar varias capas de la aplicación con el fin de separar las diferentes partes del proyecto.



Figura 5-1 Capas utilizadas en la implementación del proyecto de tesis

5.3 Proceso de implementación del sistema

La implementación del sistema inició con la codificación de la aplicación servidor. Se implementaron las clases básicas de búsqueda de recomendaciones. Primeramente se creó el servicio Web que exponía los métodos que iban a ser consumidos por la extensión. Se empezó con un método que obtenía las entidades provenientes del servicio Web de ClearForest. Una vez que se pudo establecer la comunicación con dicho servicio Web, se procedió a parsear el resultado y encapsularlo en objetos denominados EntityInfo, con el fin de obtener la información relevante para nosotros de las entidades encontradas por ClearForest en los archivos HTML que enviamos para su análisis.

Dado que el proyecto está desarrollado bajo una arquitectura de programación en capas, se creó un proyecto de librería de clases en Visual Studio con el fin de que albergue todas aquellas clases que sirven de comunicación entre las diferentes capas de la aplicación.

Luego de que se pudo obtener las entidades correspondientes, se procedió a implementar la funcionalidad de obtención de recomendaciones de música, para lo cual se utilizó una librería de clases de código abierto de Last.fm desarrollada para .Net, que permitió la implementación de métodos para buscar pistas en Last.fm según varios criterios como son artista y nombre de canción.

Luego se implementó la aplicación cliente representada por la extensión en Mozilla Firefox. Se procedió a crear la interfaz de interacción con el usuario. Se crearon archivos XUL que representaban el área de notificación en la barra de estado, la barra lateral destinada a mostrar las recomendaciones generadas en el día y las opciones del menú que permitan configurar la aplicación.

La funcionalidad en la extensión fue implementada utilizando Javascript mediante funciones que hacían llamadas a componentes expuestos a

través de una interfaz IDL y creados para cumplir con los requerimientos del sistema, entre ellos establecer la comunicación con el servicio Web de Reommedia, parsear y actualizar archivos XML, entre otros.

Mientras se desarrollaba la extensión se continuó con la implementación de la funcionalidad restante del servicio. Se crearon las clases correspondientes para almacenar información de vídeos y sitios Web, además fue necesario crear clases encargadas de buscar la información en YouTube y Del.icio.us, respectivamente.

Una vez que se pudo mostrar las recomendaciones en Mozilla Firefox a través de la extensión, se procedió a desarrollar la parte de almacenamiento en la base de datos. Se crearon las clases necesarias para habilitar el acceso y administración de los datos. Luego fue necesario realizar un filtrado de las recomendaciones para evitar enviar al usuario sugerencias repetidas.

Luego de que se realizó el almacenamiento de las recomendaciones en la base de datos, continuamos con la implementación de la funcionalidad necesaria para que el usuario califique las recomendaciones recibidas, según sus gustos. Se implementó la

interfaz necesaria en la extensión de Firefox y luego se desarrolló la funcionalidad en el servicio Web.

Finalmente, una vez desarrollada la funcionalidad necesaria para calificar las recomendaciones recibidas, decidimos utilizar las recomendaciones con mayor calificación otorgada por el usuario, para generar nuevas recomendaciones.

5.4 Requerimientos de hardware y software

A continuación se detallan los requerimientos de software necesarios para la ejecución de los componentes de Recommedia.

Aplicación cliente:

- Mozilla Firefox versión 2.X.X.X.
- Extensión Slogger.

Aplicación servidor:

- Sistema Operativo Windows XP/Vista/Server 2003.
- Microsoft SQL Server 2005 con Service Pack 1.
- Internet Information Services versión 5 en adelante.

- Microsoft .Net Framework 2.0.

Los requerimientos de hardware necesarios para que el sistema pueda ser ejecutado son una PC de escritorio y conexión a internet.

La instalación de la aplicación cliente debe ser realizada en el siguiente orden:

1. **Extensión Slogger.** Una vez instalada, esta extensión debe ser configurada de la siguiente manera:
2. Ir a *Slogger Settings* ubicado en el menú Herramientas → Slogger -> Settings de Mozilla Firefox. Aparecerá la siguiente pantalla:

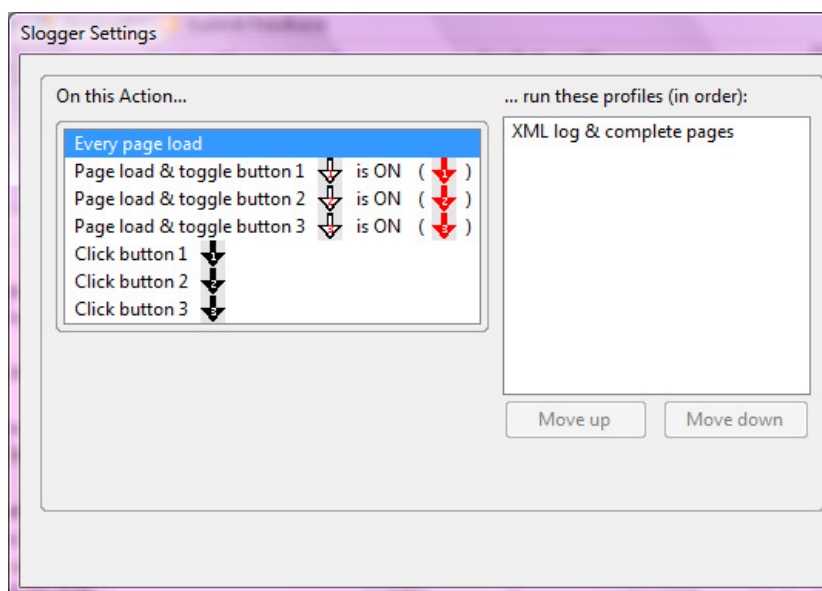


Figura 5-2 Ventana de configuración de Slogger

3. En la sección *Available Profiles*, seleccionar la opción XML log & complete pages y presionar el botón “Edit Profile”.

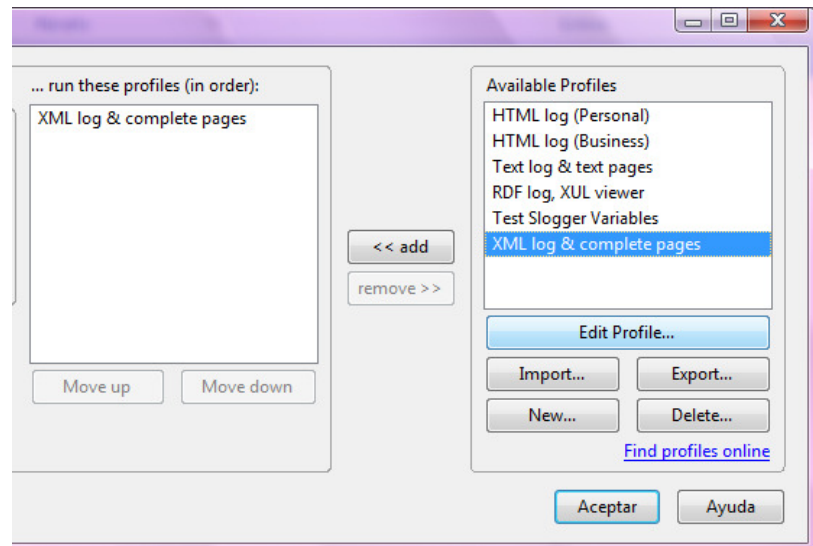


Figura 5-3 Sección Available Profiles de Slogger

4. En la pestaña *Save Pages*, seleccionar la opción Enable save pages. Seleccionar también la opción Save each page as: Web page, HTML only.

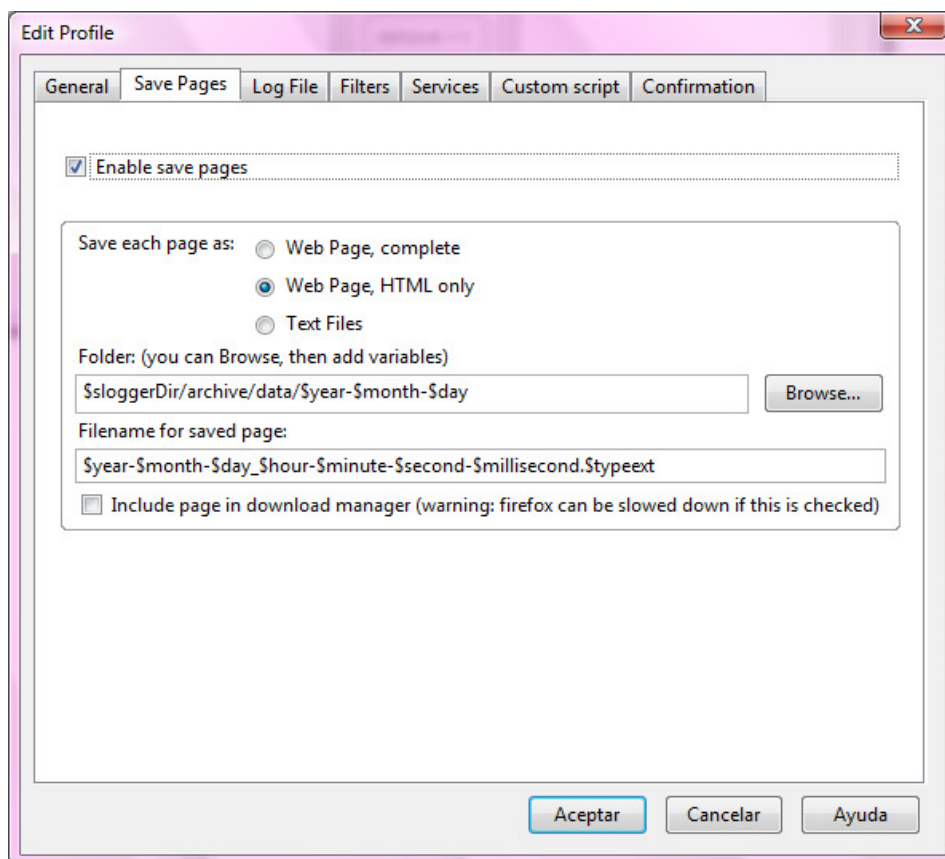


Figura 5-4 Ventana de edición de perfiles de Slogger

5. Extensión Recommedia. Esta extensión no requiere configuración adicional alguna, una vez instalada.

5.5 Problemas de implementación

Dentro del proceso completo de desarrollo del sistema surgieron algunos problemas que fueron solucionados analizando alternativas y la investigación en la Web.

Durante el desarrollo se necesitaba monitorear el directorio de archivos resultantes del historial de navegación del usuario, generados por Slogger. Después de analizar y probar posibles soluciones al problema, se resolvió que utilizando XPCOM y varios componentes que permiten la manipulación de archivos y de directorios se podría obtener los archivos necesarios.

La implementación también incluye establecer comunicación de la extensión con el servicio Web, para lo cual se empezó utilizando la comunicación por SOAP a través de XPCOM, pero no se obtenía respuesta del servicio. Luego, después de consultas online con expertos en desarrollo en Mozilla, acordamos utilizar HTTP para acceder al servicio Web, lo cual funcionó de manera adecuada.

Una vez obtenidas las recomendaciones, se requería que estas se muestren en una barra lateral, en donde se pueda poner la calificación buena o mala en dos columnas, y que el usuario pueda seleccionar una celda específica dependiendo de su opinión sobre determinada recomendación. Esto no se pudo lograr realizar pues Gecko, la plataforma sobre la que trabaja Mozilla, no soporta realizar este tipo de interfaces en donde en un control de tipo lista se pueda seleccionar las celdas individualmente.

Finalmente, al desarrollar la aplicación servidor aparecieron dos problemas: establecer la forma de obtener las recomendaciones de sitios Web y vídeos desde Del.icio.us y YouTube, respectivamente; y errores de parseo en los archivos XML de las recomendaciones generadas.

El primer problema consistía en implementar el mecanismo para generar las recomendaciones para sitios Web y para vídeos. Una de las alternativas que se encontraron fue la de utilizar API's disponibles tanto para del.icio.us y YouTube, pero estas API's no eran lo suficientemente configurables, pues no contenían los métodos necesarios para obtener las recomendaciones de acuerdo a las entidades recopiladas del historial de navegación del usuario. La segunda opción y la que representó la solución al problema fue utilizar RSS de los sitios Web de Del.icio.us y YouTube, porque permite realizar búsquedas por palabras claves, en nuestro caso entidades.

El segundo problema encontrado en la implementación del servidor estuvo en la aparición de errores de parseo del archivo XML de las recomendaciones. El origen de este problema estaba en que algunos de los resultados de las búsquedas obtenidas mediante RSS incluían caracteres latinos como las vocales tildadas y otros símbolos especiales tales como paréntesis, numeral o arroba que no eran correctamente

interpretados por el navegador Web al momento de mostrar la lista de recomendaciones. El problema fue resuelto creando una función que convertía este tipo de caracteres a su equivalente en la codificación HTML respectiva.

5.6 Conclusiones

En este capítulo se presentaron los estándares de implementación del sistema, los requerimientos de hardware y software necesarios para su instalación y los problemas presentados en esta etapa.

Para el desarrollo de los componentes de software correspondientes al servicio Web, se seleccionó la plataforma .NET y Microsoft SQL Server 2005 para la instalación de la base de datos. En el desarrollo de la extensión, se utilizaron los lenguajes con los que interactúa Mozilla Firefox: XUL y JavaScript.

Los problemas de monitoreo de directorios y archivos en la aplicación cliente fueron resueltos con el uso de XPCOM. Para generar recomendaciones de vídeos y sitios Web en el servidor utilizamos RSS. Y para los problemas entre la extensión y el servicio Web la solución fue utilizar HTTP como canal de comunicación.

CAPÍTULO 6

PRUEBAS

En este capítulo se hace una descripción de las pruebas efectuadas al sistema para verificar su funcionamiento, usabilidad y medir la efectividad de las recomendaciones que el mismo genera. Además se muestra un análisis de los resultados de las pruebas del sistema.

6.1 Esquema de pruebas

En esta sección se hace una descripción general de las estrategias de pruebas realizadas al sistema para garantizar su correcto funcionamiento.

Se solicitó a cinco personas seguir una serie de tareas específicas, consideradas claves para las pruebas. Los perfiles de las personas que participaron en este proceso de pruebas fueron los siguientes: Dos profesionales en Ingeniería en Computación, por lo tanto, son personas con conocimientos altos de manejo de computadoras y navegación en internet. Además, solicitamos a dos personas con conocimientos medios de internet y del uso de computador, que participen en las pruebas, junto con una persona con los conocimientos básicos en computadoras y poca experiencia de navegación en la web. Este experimento tuvo el objeto de probar qué tan eficaz era el sistema para el cumplimiento de dichas tareas. La ficha de la prueba puede revisarse en el anexo A.

Objetivos:

- Verificar que la interfaz y los mensajes que el sistema genera sean comprendidos por el usuario.

- Medir la efectividad de la recomendación que realiza el sistema.

¿Qué se quiere capturar?:

- Que la interfaz del sistema sea comprendida sin dificultades por el usuario y que pueda interactuar con ella de manera espontánea.
- Que las recomendaciones generadas por el sistema estén relacionadas a las preferencias de navegación del usuario.
- Que el contenido de las recomendaciones generadas se adapte a las preferencias en la navegación del usuario. Es decir que el sistema sea adaptivo.

Grupo Objetivo: Cinco personas con conocimiento sobre navegación en Internet.

Procedimiento:

El procedimiento para realizar las pruebas del sistema está dividido en dos etapas. La primera, es una etapa de evaluación de la usabilidad del sistema.

Se entregará a cada persona una hoja con 3 pasos detallados relacionados a la interacción con el sistema, seguidos de un pequeño

cuestionario para evaluar la dificultad en la realización de esta parte de la prueba.

La segunda etapa cumple el objetivo de medir la efectividad de las recomendaciones que genera el sistema, así como también la adaptabilidad del sistema a cambios en los intereses de los usuarios.

Prueba de generación de recomendaciones

Se pedirá a cada persona que navegue libremente por sitios de su interés por 15 minutos. Se prevé que en el transcurso de este tiempo las notificaciones del sistema se comiencen a mostrar. Se indicará a cada persona que revise el listado de recomendaciones ubicado en la barra lateral cuando aparezca la primera notificación en su navegador. Cada vez que el usuario desee, puede revisar ese listado sin importar si una nueva notificación ha aparecido en el navegador. En esta parte las personas pueden guiarse con el procedimiento de la Etapa 1 en caso de no recordar cómo hacerlo. Una vez terminados los 15 minutos de navegación, se entregará un cuestionario y se solicitará a las personas que revisen las recomendaciones generadas y que basados en el contenido de dichas recomendaciones contesten las preguntas del cuestionario (cuestionario 2 del anexo A). Se les consultará sobre la utilidad de las recomendaciones generadas.

Prueba de adaptabilidad del sistema

Luego se solicitará a las personas que cambien completamente sus tópicos de navegación y los mantengan durante 15 minutos más. De la misma manera que en la prueba anterior, cuando las notificaciones comiencen a mostrarse las personas están en total libertad de revisar las recomendaciones en el momento que deseen. Al finalizar el tiempo se entregará un cuestionario (cuestionario 3 del anexo A) en el cual se consultará sobre la adaptabilidad del sistema, si las recomendaciones están relacionadas a los nuevos temas de navegación.

6.2 Análisis de resultados

Los usuarios pudieron llevar a cabo las tareas requeridas y de acuerdo a las respuestas obtenidas en el cuestionario 1, que abarca la usabilidad del sistema, concluimos que el 93.34% de los usuarios participantes en las pruebas considera que el sistema es fácil o muy fácil de utilizar. La tarea que representó un pequeño grado de dificultad, fue aquella en la que se pedía al usuario ver las recomendaciones, puesto que se debe acceder a esta opción utilizando el menú contextual del sistema. El 80% de los participantes puede revisar las recomendaciones en la barra lateral sin ningún problema y el 93% puede recordar como interactuar con el sistema.

En general, estos resultados nos indican que la interfaz ofrecida por el sistema les permite a los usuarios realizar las tareas requeridas sin problemas.

Los resultados de la segunda etapa del procedimiento de las pruebas indican que el 60% de los usuarios cree que las recomendaciones son útiles. Mientras que el mismo porcentaje de las personas evaluadas consideran que el sistema se adapta a sus preferencias de navegación.

Por otro lado, el 40% de las respuestas indican que el sistema genera recomendaciones repetidas, lo que nos indica que ciertos aspectos de control deben ser mejorados en el sistema.

Un reporte completo de los resultados de las pruebas pueden encontrarse en el Anexo B.

Concluimos que las pruebas efectuadas enfocadas hacia el usuario común revelaron que, de manera general, el sistema cumple con los objetivos planteados al inicio de este capítulo, pero no en el 100% de los casos. Por consiguiente, queda pendiente pulir aspectos para el mejoramiento del sistema tales como mejorar el procedimiento de

generar las recomendaciones y control de recomendaciones ya generadas.

CONCLUSIONES Y RECOMENDACIONES

CONCLUSIONES

Una vez finalizado el presente trabajo de tesis y llevado a cabo todos los objetivos de su realización, se puede llegar a las siguientes conclusiones:

1. La gran cantidad de información existente en Internet, hace necesario el uso de técnicas que ayuden al usuario a encontrar lo que desea. Los sistemas de recomendación son utilizados eficientemente para resolver el problema de sobrecarga de información.
2. Todos los objetivos planteados respecto al análisis y diseño previos a la implementación de esta solución se cumplieron, y ello se demuestra en la documentación aquí presentada.
3. La solución ofrecida permite recibir recomendaciones de música, vídeos y sitios Web, utilizando el navegador Web Mozilla Firefox, puesto que el sistema le brinda una interfaz de notificación de recomendaciones.
4. De acuerdo a la forma en que el sistema fue diseñado, éste se puede adaptar a nuevos tipos de recomendaciones, de tal manera que no incluya solo los tres tipos de recomendaciones que se plantearon como objetivo en esta tesis, sino que sea expandible a nuevos tópicos de interés de los usuarios.

5. Es posible construir un sistema de recomendación de información para el usuario final que se adapte a los cambios en sus intereses, los cuales son reflejados en su historial de navegación.
6. El sistema de recomendación fue implementado utilizando servicios Web, lo que facilita su uso posterior a través de diferentes tipos de interfaz, como pueden ser una aplicación Web o una aplicación de escritorio.

RECOMENDACIONES

A continuación se detallan las recomendaciones que deberían tomarse en consideración:

1. Ya que en la actualidad se ha despertado gran interés por la generación de recomendaciones que capten la atención de los usuarios, sobretodo en el ámbito comercial, herramientas como la desarrollada en esta tesis serán de utilidad como base para el desarrollo de sistemas más eficientes que generen mejores recomendaciones en diferentes tópicos, como por ejemplo, recomendar personas, y no sólo se limiten a vídeos, música o sitios Web.
2. Permitir la personalización de las opciones de la aplicación, tales como la cantidad de recomendaciones que se desea recibir por día, con el fin de que el usuario no se sienta saturado de recomendaciones.
3. Convertir al proceso de recomendar en una experiencia social, de manera que los usuarios puedan interactuar entre ellos y no sólo interactúen con el sistema.
4. Luego de basar nuestra tesis en sistemas de recomendación, encontramos interesante aplicar dicho concepto no sólo en la Web, sino trasladarlo a otros entornos como lo es la televisión, aparatos

eléctricos de hogar o dispositivos móviles. Podemos pensar que en un futuro podríamos tener a nuestro refrigerador recomendándonos que comer para desayunar.


5. Al generar recomendaciones, en ciertas ocasiones es importante conocer no solamente qué recomendar sino también a quién y bajo qué circunstancias. Es decir, es significativo conocer la información contextual del usuario para generar recomendaciones más acertadas. En el presente trabajo no se ha utilizado información contextual de los usuarios.
6. Es posible mejorar la forma de generar las recomendaciones para que sean más útiles al usuario y se adapten mejor a sus preferencias, esto se puede lograr haciendo un análisis más exhaustivo de la navegación del usuario, como por ejemplo llevando el registro del tiempo que visita una página.
7. Modificar el mecanismo de filtrado para evitar la generación de recomendaciones repetidas. El sistema en estos momentos cuenta con un procedimiento de filtrado que se basa en las URLs, consideramos que se podría basar también en las entidades que se obtienen de la navegación de los usuarios, descartándolas en el caso de que antes ya hayan sido enviadas al servidor.

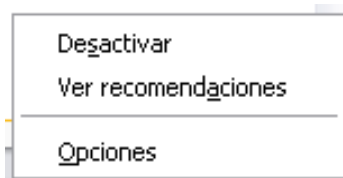
ANEXOS

Anexo A

Procedimiento para pruebas del sistema

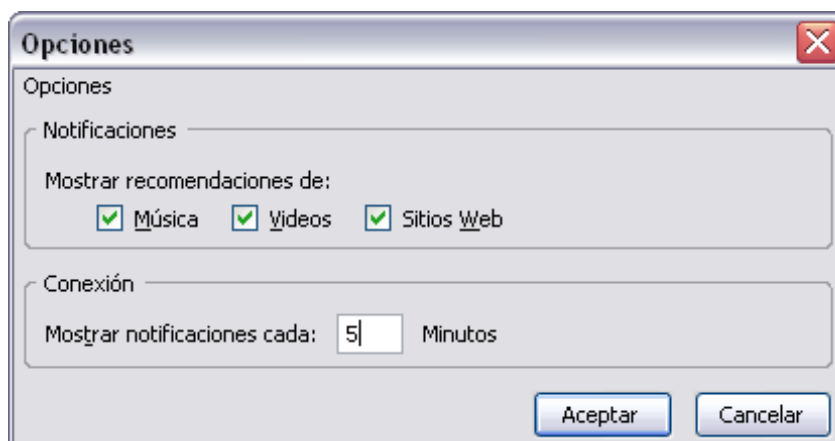
Etapa 1

1. En la ventana del navegador haga clic derecho en el ícono de la extensión en la barra de estado (ubicada en la parte inferior derecha) identificado con la imagen . Se aprecia el siguiente menú contextual



Menú contextual de Recommedia

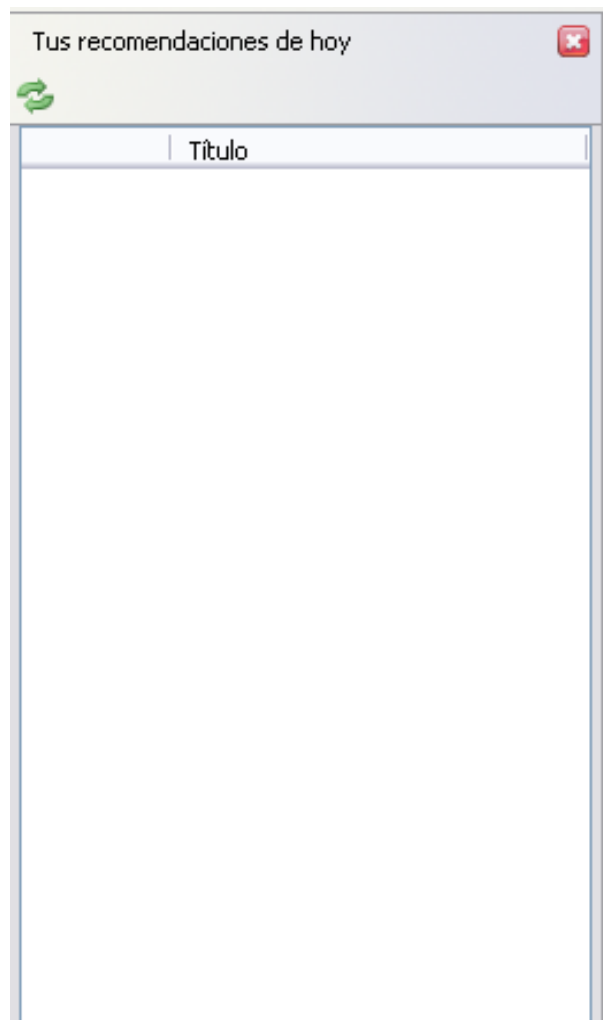
2. Del menú contextual desplegado en el paso 1 escoja *Opciones*. Aparece la siguiente ventana:






Ventana de Opciones de Recommedia

Aquí puede fijar dos parámetros de configuración de la extensión, el tipo de recomendaciones a ser mostradas, y la frecuencia con la que se mostrarán las notificaciones. Configure estos valores de acuerdo a sus preferencias y haga clic en *Aceptar*.

3. Del menú contextual desplegado en el paso 1 escoja *Ver recomendaciones*. Se abrirá una barra lateral en la parte izquierda del navegador:



Barra Lateral donde se muestran las recomendaciones

Por ser la primera vez que utiliza la extensión, esta barra lateral lucirá vacía. Pero una vez que las recomendaciones comiencen a generarse aparecerán los diferentes ítems que podrán ser reconocidos por sus iconos    para recomendaciones de vídeos, música y sitios Web respectivamente.

Cuestionario 1

Marque con una X su respuesta.

1. Evalúe el nivel de dificultad de cada tarea marcando en la casilla la respuesta que corresponda.

Nivel de Dificultad	Tarea 1	Tarea 2	Tarea 3
Muy fácil			
Fácil			
Difícil			
Muy difícil			

2. ¿Necesitó ayuda para realizar la tarea? (aquí se puede medir el número de errores).

Tarea	Si	¿Cuál fue la ayuda?	No
1			
2			
3			

3. ¿Puede recordar las tareas y repetirlas sin necesidad de consultar el procedimiento entregado a usted en la tarea anterior?

Tarea	Si	No	¿Por qué?
1			
2			
3			

Etapa 2

Cuestionario 2

Conteste las siguientes preguntas relacionadas a las recomendaciones que obtuvo luego de navegar por 15 minutos.

1. ¿Las recomendaciones han sido útiles para usted?

SI _____ NO _____

2. ¿Las recomendaciones han sido interesantes para usted?

SI _____ NO _____

3. ¿Ha recibido alguna recomendación fortuita o que le cause impresión?

SI _____ NO _____

4. ¿Las recomendaciones estaban relacionadas a los temas que usted eligió para su navegación?

SI _____ NO _____

Cuestionario 3

Conteste las siguientes preguntas relacionadas a las recomendaciones que obtuvo luego de navegar por los segundos 15 minutos.

1. ¿Las nuevas recomendaciones están relacionadas a los temas de su navegación durante los 15 primeros minutos?

SI _____ NO _____

2. ¿Percibe un cambio en los resultados de las recomendaciones generadas entre la primera parte de esta prueba y los últimos 15 minutos?

SI _____ NO _____

3. ¿Ha encontrado recomendaciones repetidas?

SI _____ NO _____

Anexo B

Resultado de Pruebas

Para cada cuestionario se presenta la tabla de respuestas correspondiente a la evaluación de las 5 personas.

Cuestionario 1

Evalúe el nivel de dificultad de cada tarea marcando en la casilla la respuesta que corresponda.

Tabla de respuestas:

Nivel de Dificultad	Tarea 1	Tarea 2	Tarea 3	Promedio
Muy fácil	5 (100%)	2 (40%)	0	46,67 %
Fácil	0	3 (60%)	4 (80%)	46,67 %
Difícil	0	0	1 (20%)	6,67%
Muy difícil	0	0	0	0%

¿Necesitó ayuda para realizar la tarea?

Tarea	Si	No
1	0	5 (100%)
2	1 (20%)	4 (80%)
3	2 (40%)	3 (60%)

¿Puede recordar las tareas y repetirlas sin necesidad de consultar el procedimiento entregado a usted?

Tarea	Si	No
1	5 (100%)	0
2	4 (80%)	1 (20%)
3	5 (100%)	0

Etapa 2

Cuestionario 2

Conteste las siguientes preguntas relacionadas a las recomendaciones que obtuvo luego de navegar por 15 minutos.

1. ¿Las recomendaciones han sido útiles para usted?

SI	NO
3 (60%)	2 (40%)

2. ¿Las recomendaciones han sido interesantes para usted?

SI	NO
2 (40%)	3 (60%)

3. ¿Ha recibido alguna recomendación fortuita o que le cause impresión?

SI	NO
1 (20%)	4 (80%)

4. ¿Las recomendaciones estaban relacionadas a los temas que usted eligió para su navegación?

SI	NO
3 (60%)	2 (40%)

Cuestionario 3

Conteste las siguientes preguntas relacionadas a las recomendaciones que obtuvo luego de navegar por los siguientes 15 minutos.

1. ¿Las nuevas recomendaciones están relacionadas a los temas de su navegación durante los 15 primeros minutos?

SI	NO
1 (20%)	4 (80%)

2. ¿Percibe un cambio en los resultados de las recomendaciones generadas entre la primera parte de esta prueba y los últimos 15 minutos?

SI	NO
4 (80%)	1 (20%)

3. ¿Ha encontrado recomendaciones repetidas?

SI	NO
2 (40%)	3 (60%)

BIBLIOGRAFÍA

- [1] Mitchell, Tom M. (Marzo 1997) *Machine Learning*. McGraw-Hill.
- [2] Chesani, Federico. *Recommendation Systems*. Obtenida en Octubre del 2007, de
<http://www-db.deis.unibo.it/courses/SI2/Relazioni/RecSystems.pdf>.
- [3] Fdez. Deleito, Olga. Diseñar Sistemas de Recomendación: Introducción. Obtenida en agosto del 2007, de
<http://www.nethodical.com/archivos/000048.html>.
- [4] R. Torres et al. (2004), Enhancing digital libraries with TechLens, *Digital Libraries, 2004. Proceedings of the 2004 Joint ACM/IEEE Conference*.

- [5] Clear Forest, Text Analytics Solutions. Obtenida en septiembre del 2007, de <http://www.clearforest.com>.
- [6] Outfoxed, Obtenida en octubre 06 del 2007, de <http://getoutfoxed.com/about>
- [7] S. M. McNee, J. Riedl, and J. A. Konstan (2006), Making recommendations better: an analytic model for human-recommender interaction, *Conference on Human Factors in Computing Systems*.
- [8] Saboya Vargas, Aniceto (2005). Uso de Recomendadores, Asistentes y Ayudantes en sistemas Tutores. Dpto. LSI. *Universidad Politécnica de Cataluña*.
- [9] Slogger, Obtenida en diciembre 1 del 2007, de <http://www.kenschutte.com/slogger/>
- [10] A. Gulli and A. Signorini (2005). The Indexable Web Is More Than 11.5 Billion Pages, *International World Wide Web Conference*.
- [11] R. Burke (2002), Hybrid Recommender Systems: Survey and Experiments, *User Modeling and User-Adapted Interaction* 12, no. 4.
- [12] Selberg, E. (1999). Towards Comprehensive Web Search. Tesis de PhD. Universidad de Washington. Obtenida de <http://citeseer.ist.psu.edu/selberg99towards.html>
- [13] World Wide Web Size. Obtenida en diciembre 16 del 2007, de <http://www.worldwidewebsize.com/>

- [14] I. Gantz et al. (Marzo 2007), The Expanding Digital Universe—A Forecast of Worldwide Information Growth Through 2010, *An IDC White Paper—sponsored by EMC*.
- [15] Kohrs, Arnd (Julio 2001), Tesis de Doctorado: Collaborative Filtering on the Internet. *Universidad de Nice Sophia Antipolis*, Francia Obtenida en Marzo 2007, de <http://www.eurecom.fr/util/publownload.en.htm?id=931>
- [16] J. Rucker and M. J. Polanco (1997), SiteSeer: Personalized Navigation for the Web, *Communications of the ACM* 40, no. 3.
- [17] Smyth, B., McCarthy, K., Reilly, J., O`Sullivan, D., McGinty, L., and Wilson, D.C. (2005) Case Studies in Association Rule Mining for Recommender Systems. In: Arabnia, H.R., and Joshua, R. (eds) *Proceedings of the 2005 International Conference on Artificial Intelligence (ICAI-05)*.
- [18] C. N. Seon et al. (2001), Named Entity Recognition Using Machine Learning Methods and Pattern-Selection Rules, *Proceedings of the Sixth Natural Language Processing Pacific Rim Symposium*.
- [19] Wikipedia, Text Analytics Obtenida en enero del 2008, de http://en.wikipedia.org/wiki/Text_analytics.
- [20] Ronen Feldman and James Sanger (2007), *The Text Mining Handbook: Advanced Approaches in Analyzing Unstructured Data*. Cambridge University Press.

- [21] M. Claypool et al. (2001), Implicit Interest Indicators, *Proceedings of the 6th international conference on Intelligent user interfaces*.
- [22] W3C, Web Semantic. Obtenida en enero del 2008, de <http://www.w3.org/2001/sw/>
- [23] P. J. Brown y G. J. F. Jones (2001), Context-Aware Retrieval: Exploring a New Environment for Information Retrieval and Information Filtering, *Personal and Ubiquitous Computing* 5, no. 4.
- [24] Ingwersen, Peter. (1992), Information Retrieval Interaction. *Taylor Graham Publishing*, London.
- [25] Allan, James., Aslam, Jay (Septiembre 2002). Challenges in Information Retrieval and Language Modeling. *Report of a Workshop held at the Center for Intelligent Information Retrieval*, University of Massachusetts Amherst.
- [26] Dilip Sheth, Beerud (Febrero 1994). A learning approach to personalized information filtering. Tesis previa a la obtención del título de Master of Science in Computer Science and Engineering. Massachusetts Institute of Technology.
- [27] Sarwar, Badrul., Karypis, George., Konstan, Joseph., Riedl, John. Item Based Collaborative Filtering Recommendation Algorithms. *GroupLens Research Group/Army HPC Research Center*. Department of Computer Science and Engineering. University of Minnesota, Minneapolis.

- [28] Belkin, Nicholas J., Croft, W. Bruce. (Diciembre 1992) Information Filtering and Information Retrieval. Two Sides of the same Coin?. *Communications of the ACM. Vol 35, No 12.*
- [29] Osorio Hincapié, Lina María., Mejía Rengifo, Edwing Andrés., Guzmán Luna, Jaime Alberto (2006). Metis: Sistema de Recomendación de Documentos Digitales basado en un Modelo Colaborativo. *Universidad Nacional de Colombia, Medellín, Colombia.*
- [30] Towle, B. and Quinn, C. (2000), Knowledge Based Recommender Systems Using Explicit User Models. *In Knowledge-Based Electronic Markets, Papers from the AAAI Workshop, AAAI Technical Report WS-00-04. pp. 74-77. Menlo Park, CA: AAAI Press.*
- [31] McNee, Sean (Junio 2006). Meeting User Information Needs in Recommender Systems. *Universidad de Minnesota. Tesis doctoral.*
- [32] W3C (Febrero 2004). Web Services Architecture. *W3C Working Group Note 11.*
- [33] Goodman, Elizabeth. , Moed, Andrea (2006). Community in Mashups: The Case of Personal Geodata. *UC Berkeley School of Information.*
- [34] McNee, Sean. , Riedl, John. , Konstan, Joseph (2006). Accurate is not always good: How Accuracy Metrics have hurt Recommender Systems. *Department of Computer Science and Engineering University of Minnesota.*

- [35] Ziegler, C.N., McNee, S.M., Konstan, J.A., Lausen, G. (2005),
Improving Recommendation Lists through Topic Diversification. In
Proc. of WWW 2005, ACM Press, 22-32.
- [36] FOAF Project. Obtenida en Julio del 2007, de
<http://www.foaf-project.org>
- [37] Tim Berners – Lee (Mayo 1994) W3 Future Directions *Keynote, 1st*
World Wide Web Conference. Geneva.
- [38] Sitio Web de Slogger. Obtenido en Julio 2007, de
<http://www.kenschutte.com/slogger/>
- [39] Hammond, Tony. Hannay, Timo, Lund, Ben y Scott, Joanna (Abril
2005). “Social Bookmarking Tools (I)”, *Nature Publishing Group. D –*
Lib Magazine, Volumen 11 Número 4.
- [40] Sitio Web de RapidMiner. Obtenida en abril del 2008, de
<http://rapid-i.com/content/blogcategory/10/69/>
- [41] XUL - Mozilla developer center. Obtenida en abril del 2008, de
<http://developer.mozilla.org/es/docs/XUL>
- [42] XPInstall - Mozilla developer center. Obtenida en abril del 2008, de
<http://developer.mozilla.org/es/docs/XPInstall>
- [43] Javascript - Mozilla developer center. Obtenida en abril del 2008, de
<http://developer.mozilla.org/es/docs/JavaScript>
- [44] DOM - Mozilla developer center. Obtenida en abril del 2008, de
<http://developer.mozilla.org/es/docs/DOM>

- [45] CSS - Mozilla developer center. Obtenida en abril del 2008, de <http://developer.mozilla.org/es/docs/CSS>
- [46] XPCOM - Mozilla developer center en español. Obtenida en abril del 2008, de <http://developer.mozilla.org/es/docs/XPCOM>
- [47] XPCOM - Mozilla developer center en inglés. Obtenida en abril del 2008, de <http://developer.mozilla.org/en/docs/XPCOM>
- [48] XPConnect Mozilla developer center. Obtenida en abril del 2008, de <http://developer.mozilla.org/en/docs/XPConnect>
- [49] Scriptable Components - Mozilla.org. Obtenida en abril del 2008, de <http://www.mozilla.org/scriptable/>
- [50] Simple Object Access Protocol (SOAP) 1.1. Obtenida en abril del 2008, de <http://www.w3.org/TR/2000/NOTE-SOAP-20000508/>
- [51] WSDL – Wikipedia Obtenida en abril del 2008, de <http://es.wikipedia.org/wiki/WSDL>
- [52] Sitio Oficial de XML-RPC Obtenida en abril del 2008, de <http://www.xmlrpc.com/>
- [53] Design Guidelines for Class Library Developers. Obtenida en septiembre del 2008, de <http://msdn.microsoft.com/en-us/library/czefa0ke.aspx>
- [54] Descubre qué palabras son las más buscadas en Internet y los temas que están en auge. Obtenida en noviembre del 2008, de <http://www.noticias.com/articulo/30-08-2004/montse->

penarroja/descubre-que-palabras-mas-buscadas-internet-y-temas-que-estan-auge-46fm.html

- [55] Comunicación entre procesos. Obtenida en diciembre del 2008, de http://es.wikipedia.org/wiki/Comunicacion_entre_procesos
- [56] Comunicación entre procesos – Introducción. Obtenida en diciembre del 2008, de <http://tvdi.det.uvigo.es/~belen/pem/apuntes/node21.html>
- [57] Mentzer, John. Moon, Mark (2004). *Sales Forecasting Management: A demand management Approach* Editorial Sage. Segunda Edición.
- [58] Jolliffe, I. T. Stephenson, David (2003). *Forecast Verification: A practitioner's guide in atmospheric science*. Editorial John Wiley and Sons.
- [59] Gonen, Mithat. Receiver Operating Characteristic (ROC) Curves. Memorial Sloan – Kettering Cancer Center.
- [60] Tape, Thomas, MD. Interpreting Diagnostic Tests. *University of Nebraska Medical Center*. Obtenida en julio del 2009, de <http://gim.unmc.edu/dxtests/Default.htm>.
- [61] Blattberg, Robert. Pyöng-do, Kim. Neslin, Scott A (2008). *Database Marketing: Analyzing and Managing Customers*. Editorial Springer.
- [62] James, Stan (Octubre 2007). Outfoxed: Trusted metadata distribution using social networks.
- [63] Baeza – Yates, R. Ribeiro – Neto, B. (1999). *Modern Information Retrieval ACM Press*; Editorial Addison – Wesley.

- [64] Propuesta y desarrollo de un modelo para la evaluación de la recuperación de la información en la Web. Obtenida en julio del 2009, de http://descargas.cervantesvirtual.com/servlet/SirveObras/02472741989036164198835/010010_6.pdf
- [65] Sarwar, Badrul. Konstan, Joseph. Borchers, Al. Herlocker, Jon. Miller, Brad. Ried, John (1998). Using Filtering Agents to Improve Prediction Quality in the GroupLens Research Collaborative Filtering System. *GroupLens Research Project y .Net Perceptions, Inc.*