



UNIVERSIDAD CENTRAL DE VENEZUELA
FACULTAD DE CIENCIAS
ESCUELA DE COMPUTACIÓN
CENTRO DE ENSEÑANZA ASISTIDA POR COMPUTADOR - CENEAC

Implementación de un Sistema de Recomendaciones de frases basado en el algoritmo *Slope One*

Trabajo Especial de Grado presentado ante la Ilustre
Universidad Central de Venezuela por el
Br. Norman Riveros Balbi (Cl. 14.891.918)
para optar al título de Licenciado en Computación

Tutora: Profa. Yusneyi Y. Carballo Barrera

Ciudad Universitaria de Caracas, 07 de Febrero de 2013

UNIVERSIDAD CENTRAL DE VENEZUELA
FACULTAD DE CIENCIAS
ESCUELA DE COMPUTACIÓN
CENTRO DE ENSEÑANZA ASISTIDA POR COMPUTADOR - CENEAC

ACTA

Quienes suscriben, miembros del jurado designado por el Consejo de la Escuela de Computación, para examinar el Trabajo Especial de Grado titulado **“Implementación de un Sistema de Recomendaciones de frases basado en el algoritmo *Slope One*”** y presentado por el Br. **Norman Riveros Balbi (C.I. V-14891918)**, a los fines de optar al título de **Licenciado en Computación**, dejamos constancia de lo siguiente:

Leído como fue dicho trabajo, por cada uno de los miembros del jurado, se fijó el día ____ de _____ de _____, a las _____ horas, para que el autor lo defendiera en forma pública, lo cual éste hizo en _____ de la Escuela de Computación, mediante una presentación oral de su contenido, luego de lo cual respondió a las preguntas formuladas. Finalizada la defensa pública del Trabajo Especial de Grado, el jurado decidió aprobar con la nota de ____ puntos.

En fe de lo cual se levanta la presente Acta, en Caracas el día ____ de _____ de _____.

Profa. Yusneyi Carballo Barrera
(Tutora)

Prof. Iván Flores
(Jurado)

Profa. Zenaida Castillo
(Jurado)

Agradecimientos y dedicatorias

Para mi madre, que hizo lo posible y lo imposible para que yo lograra llegar a este momento.

Para mi novia, que me apoyó y supo consentirme en los momentos difíciles y arreararme en los fáciles.

Para mis amigos, que son también mi familia, gracias por su apoyo a lo largo de todos estos años.

Para mi tutora, sin la cual este proyecto nunca se hubiese realizado exitosamente.

Universidad Central de Venezuela
Facultad de Ciencias
Escuela de Computación
Centro de Enseñanza Asistida por Computador – CENEAC

Implementación de un Sistema de Recomendaciones de frases basado en el algoritmo *Slope One*

Autor: Norman Alejandro Riveros Balbi
Norman.riveros@gmail.com

Tutora: Profa. Yusneyi Carballo Barrera
yusneyi.carballo@ciens.ucv.ve

Fecha: <07/02/2013>

RESUMEN

Los sistemas de recomendación son herramientas que intentan proveer sugerencias a un usuario basándose en sus preferencias. Este trabajo describe el proceso realizado para el diseño e implementación de un sistema de recomendación de frases, basado en el algoritmo *Slope One*, y un sitio web de apoyo al sistema de recomendación. Se incluye también una investigación sobre los sistemas de recomendación y el proceso mediante el cual se seleccionó el algoritmo *Slope One* para la solución del problema y una descripción de la metodología utilizada, basada en el desarrollo incremental e iterativo. Los resultados obtenidos de dos fases de prueba realizadas indican que el sistema puede generar recomendaciones válidas a un gran porcentaje de los usuarios y que su efectividad, siendo considerada adecuada, crecerá a medida que aumenten el número de usuarios del sistema y las calificaciones realizadas por los mismos.

Palabras Claves: sistemas de recomendación, filtrado colaborativo, *Slope One*.

Índice

INTRODUCCIÓN	1
CAPÍTULO I: PLANTEAMIENTO DEL PROBLEMA.....	2
I.1 PROBLEMA DE INVESTIGACIÓN	2
I.2 JUSTIFICACIÓN DEL PROBLEMA	2
I.3 OBJETIVO GENERAL DEL TRABAJO ESPECIAL DE GRADO	3
I.4 OBJETIVOS ESPECÍFICOS	3
I.5 ALCANCE	3
I.6 LIMITACIONES.....	4
CAPÍTULO II: MARCO TEÓRICO CONCEPTUAL	5
II.1 SISTEMAS DE RECOMENDACIÓN	5
II.1.1 Definición.....	5
II.1.2 Antecedentes	6
II.1.3 Funcionalidades de los sistemas de recomendación	6
II.1.4 Datos y fuentes de conocimiento.....	8
II.1.5 Paradigmas de recomendación	10
II.1.6 Evaluación de Sistemas de Recomendación	16
II.1.7 Lineamientos para la elección de técnicas específicas	22
II.2 ALGORITMO DE PREDICCIÓN <i>SLOPE ONE</i>	30
II.2.1 <i>Slope One</i> básico	32
II.2.2 <i>Slope One</i> ponderado.....	33
II.2.3 <i>Slope One</i> bipolar.....	33
II.2.4 Resultados Experimentales.....	34
II.2.5 Algoritmo de predicción	35
II.2.6 Implementaciones.....	36

II.3 INTEGRACIÓN CON SITIOS DE REDES SOCIALES	37
II.3.1 Definición.....	37
II.3.2 Características	38
II.4 CONCLUSIONES	40
CAPÍTULO III: MARCO METODOLÓGICO	41
III.1 DESARROLLO INCREMENTAL E ITERATIVO	41
III.1.1 Etapas.....	42
III.1.2 Ventajas.....	42
III.1.3 Desventajas	43
III. 1.5 Justificación.....	44
III.2 METODOLOGÍA A SEGUIR PARA EL DESARROLLO DE LA SOLUCIÓN.....	44
III.2.1 Artefactos	44
III.2.2 Justificación	47
CAPÍTULO IV: DESARROLLO DE LA APLICACIÓN	- 48 -
IV.1 OBJETIVO GENERAL DE LA APLICACIÓN	- 48 -
IV.2 OBJETIVOS ESPECÍFICOS DE LA APLICACIÓN	- 48 -
IV.3 ALCANCE	- 48 -
IV.4 PLATAFORMA	- 49 -
IV.5 DESCRIPCIÓN DE LA APLICACIÓN.....	- 49 -
IV.5.1 Usuarios.....	- 49 -
IV.5.2 Requerimientos.....	- 50 -
IV.5.3 Funcionalidades	- 50 -
IV.5.4 Interfaces principales	- 58 -
IV.5.5 Historias de usuario principales.....	- 68 -
IV.5.6 Pruebas.....	- 71 -
RESULTADOS DEL TRABAJO ESPECIAL DE GRADO	- 74 -
CONCLUSIONES Y RECOMENDACIONES	- 75 -
REFERENCIAS BIBLIOGRÁFICAS Y DÍGITALES	- 76 -

ANEXO A – INTERFACES DEL MÓDULO DE ADMINISTRACIÓN.....	- 78 -
ANEXO B – RESULTADOS DE LAS PRUEBAS	- 80 -

Índice de figuras

FIGURA 1: ECUACIÓN PARA CALCULAR EL COEFICIENTE DE CORRELACIÓN DE PEARSON ENTRE DOS ÍTEMS	13
FIGURA 2: ECUACIÓN PARA CALCULAR LA SIMILITUD POR COSENO AJUSTADO ENTRE DOS ÍTEMS	13
FIGURA 3: ECUACIÓN PARA CALCULAR LA SIMILITUD POR RANGO DE CORRELACIÓN DE SPEARMAN ENTRE DOS ÍTEMS	14
FIGURA 4: ECUACIÓN PARA CALCULAR LA MEDIA CUADRÁTICA DEL ERROR	17
FIGURA 5: ECUACIÓN PARA CALCULAR LA MEDIA ABSOLUTA DEL ERROR	17
FIGURA 6: CLASIFICACIÓN DE LOS POSIBLES RESULTADOS DE LA RECOMENDACIÓN DE UN ÍTEM A UN USUARIO ..	18
FIGURA 7: FÓRMULAS DE MÉTRICAS DE PRECISIÓN DE PREDICCIONES DE USO	18
FIGURA 8: ECUACIÓN PARA EL CÁLCULO DEL ÍNDICE GINI	19
FIGURA 9: FÓRMULA PARA CALCULAR LA ENTROPÍA DE UN ÍTEM	20
FIGURA 10: TAXONOMÍA DE FUENTES DE CONOCIMIENTO EN RECOMENDACIÓN	23
FIGURA 11: FUENTES DE CONOCIMIENTO Y TIPOS DE RECOMENDACIÓN	24
FIGURA 12: BASE DE LOS PLANES SLOPE ONE: LAS CALIFICACIONES DEL USUARIO A DE DOS ÍTEMS Y DEL USUARIO B DE UN ÍTEM COMÚN SE UTILIZA PARA PREDECIR LA CALIFICACIÓN DESCONOCIDA DEL USUARIO B	31
FIGURA 13: FÓRMULA PARA CALCULAR LA DESVIACIÓN MEDIA DEL ÍTEM I CON RESPECTO AL ÍTEM J	32
FIGURA 14: FÓRMULA PARA CALCULAR UNA PREDICCIÓN DE CALIFICACIÓN DEL USUARIO U PARA EL ÍTEM J USANDO SLOPE ONE	33
FIGURA 15: FÓRMULA PARA CALCULAR UNA PREDICCIÓN DE CLASIFICACIÓN DEL USUARIO U PARA EL ÍTEM J USANDO SLOPE ONE PONDERADO	33
FIGURA 16: CÁLCULO DE LA DESVIACIÓN MEDIA DEL ÍTEM I CON RESPECTO AL ÍTEM J PARA ÍTEMS AGRADADOS	34
FIGURA 17: FÓRMULA PARA CALCULAR UNA PREDICCIÓN DE CLASIFICACIÓN DEL USUARIO U PARA EL ÍTEM J USANDO SLOPE ONE BIPOLAR PONDERADO	34
FIGURA 18: CICLO DE VIDA DEL DESARROLLO INCREMENTAL	42
FIGURA 19: DIAGRAMA DE ENTIDAD-RELACIÓN EXTENDIDO DE LA BASE DE DATOS.....	46
FIGURA 20: INTERFAZ DE LA PÁGINA PRINCIPAL.	- 58 -
FIGURA 21: DIÁLOGO PARA AUTENTICACIÓN DE USUARIO.....	- 58 -
FIGURA 22: DIÁLOGO PARA EL REGISTRO DE UN NUEVO USUARIO.....	- 59 -
FIGURA 23: VISTA DE LA CABECERA PARA UN USUARIO REGISTRADO EN LA APLICACIÓN.	- 59 -
FIGURA 24: VISTA DE LA CABECERA PARA UN USUARIO REGISTRADO MEDIANTE FACEBOOK.....	- 59 -
FIGURA 25: VISTA DE UNA FRASE.	- 60 -
FIGURA 26: LISTA INICIAL DE AUTORES.....	- 61 -
FIGURA 27: INTERFAZ QUE PERMITE VISUALIZAR UN AUTOR.	- 61 -
FIGURA 28: EXTRACTO DE INTERFAZ DE LISTA DE CATEGORÍAS.	- 62 -
FIGURA 29: INTERFAZ DE LA CATEGORÍA ÉXITO.	- 62 -
FIGURA 30: INTERFAZ DE LA LISTA DE FRASES FAVORITAS DEL USUARIO.....	- 63 -
FIGURA 31: INTERFAZ PARA PUBLICAR UN NUEVO ÍTEM.	- 63 -

FIGURA 32: FORMA PARA BÚSQUEDA AVANZADA DE FRASES.	- 64 -
FIGURA 33: RESULTADOS DE BÚSQUEDA AVANZADA DE FRASES.	- 64 -
FIGURA 34: RECUADRO DE BÚSQUEDA <i>FULLTEXT</i>	- 65 -
FIGURA 35: RESULTADOS DE LA BÚSQUEDA <i>FULLTEXT</i> , VISTA DE AUTORES.	- 65 -
FIGURA 36: RESULTADOS DE LA BÚSQUEDA <i>FULLTEXT</i> , VISTA DE FRASES.	- 65 -
FIGURA 37: INTERFAZ QUE PRESENTA AL USUARIO UNA LISTA DE ÍTEMS RECOMENDADOS.	- 66 -
FIGURA 38: INTERFAZ DEL ADMINISTRADOR PARA PROCESAR LOS REPORTES.	- 66 -
FIGURA 39: DIÁLOGO REPORTAR FRASE.	- 67 -
FIGURA 40: LISTA DE FRASES DEL SISTEMA.	- 67 -
FIGURA 41: FORMA PARA MODIFICAR LOS DATOS DE UNA FRASE.	- 68 -
FIGURA 42: GRÁFICO DE BARRAS REPRESENTATIVO DEL PROMEDIO DE PREDICCIONES Y EL PROMEDIO DEL VALOR ABSOLUTO DEL ERROR PARA CADA USUARIO.	- 72 -
FIGURA 43: VALORES DE LAS MÉTRICAS PRECISIÓN, MCE, MAE Y COBERTURA EN LAS FASES DE PRUEBA.	- 73 -
FIGURA 44: LISTA DE AUTORES DEL SISTEMA.	- 78 -
FIGURA 45: FORMA PARA MODIFICAR LOS DATOS DE UN AUTOR.	- 78 -
FIGURA 46: LISTA DE ETIQUETAS DEL SISTEMA.	- 79 -
FIGURA 47: LISTA DE USUARIOS DEL SISTEMA (ALGUNOS DATOS OCULTADOS MEDIANTE ASTERISCOS POR PRIVACIDAD)	- 79 -
FIGURA 48: FORMA PARA MODIFICAR LOS DATOS DE UN USUARIO.	- 79 -
FIGURA 49: GRÁFICO DE TORTAS DEL PORCENTAJE DE VERDADEROS Y FALSOS POSITIVOS EN LA PRIMERA FASE DE PRUEBA.	- 80 -
FIGURA 50: GRÁFICO DE TORTAS DEL PORCENTAJE DE VERDADEROS Y FALSOS POSITIVOS EN LA SEGUNDA FASE DE PRUEBA.	- 81 -
FIGURA 51: GRÁFICO DE TORTAS DEL PORCENTAJE DE VERDADEROS Y FALSOS POSITIVOS EN LA FASE DE PRUEBAS.	- 81 -
FIGURA 52: ERROR PROMEDIO PARA LOS USUARIOS PARTICIPANTES EN LAS 2 FASES, SEPARADO POR FASE.	- 82 -

Índice de Tablas

TABLA 1: IMPACTO DE DOMINIOS DE RECOMENDACIÓN EN FUENTES DE CONOCIMIENTO.	28
TABLA 2: FACTORES DEL DOMINIO Y TÉCNICAS DE RECOMENDACIÓN	29
TABLA 3: MAE DE TODOS MENOS UNO PARA LOS CONJUNTOS DE DATOS EACHMOVIE Y MOVIELENS DE CADA ESQUEMA, MENOR ES MEJOR.	35
TABLA 4: MATRIZ DE REQUERIMIENTOS FUNCIONALES DE LA APLICACIÓN.	44
TABLA 5: TABLA RESUMEN DE LOS RESULTADOS DE LA PRIMERA FASE DE PRUEBAS.	- 83 -
TABLA 6: TABLA RESUMEN DE LOS RESULTADOS DE LA SEGUNDA FASE DE PRUEBAS.	- 83 -
TABLA 7: TABLA RESUMEN DE LOS RESULTADOS DE AMBAS FASES DE PRUEBA.	- 83 -

INTRODUCCIÓN

A medida que la Web sigue creciendo exponencialmente, también crece el tamaño y la complejidad de muchos sitios web. Para los usuarios de estas aplicaciones es cada vez más difícil encontrar la información que buscan de forma rápida. La interfaz de usuario de un sitio web puede ayudar a sus usuarios a encontrar la información acorde a sus intereses personalizando el sitio web. En algunos casos, se presenta al usuario información personalizada permitiéndole escoger de un conjunto de tópicos de interés predefinido. Sin embargo, los usuarios no siempre saben en que están interesados con antelación y sus intereses pueden cambiar a lo largo del tiempo, lo que requeriría que cambiasen su selección frecuentemente. Los sistemas de recomendación son herramientas que intentan asistir a los usuarios en sus procesos de búsqueda de información, ayudando a filtrar los ítems encontrados utilizando recomendaciones propuestas sobre dichos ítems.

El desarrollo de sistemas de recomendación es un esfuerzo multidisciplinario que involucra saberes de varios campos como inteligencia artificial, interacción humano-computador, tecnologías de información, minería de datos, estadística, interfaces de usuario adaptables, sistemas de soporte de decisiones, mercadeo y comportamiento del consumidor. En este trabajo se presentaran los conceptos necesarios para analizar, diseñar y evaluar sistemas de recomendación, diferentes paradigmas de recomendación y varios lineamientos propuestos para la selección de una técnica de recomendación específica.

Este documento de Trabajo Especial de Grado está estructurado de la siguiente manera:

Capítulo I: Planteamiento del problema de investigación, especificando el objetivo general y los objetivos específicos del Trabajo Especial de Grado, junto con el alcance y las limitaciones.

Capítulo II: Marco teórico, que incluye la investigación de los conceptos relacionados a sistemas de recomendación, los lineamientos utilizados para la escogencia de los mismos y el algoritmo de predicción *Slope One*.

Capítulo III: Marco metodológico, consistiendo la metodología utilizada para el desarrollo de la aplicación, basada en el modelo iterativo incremental.

Capítulo IV: Desarrollo de la aplicación, detallando los objetivos generales y específicos de la aplicación, su alcance, la plataforma de implementación y una descripción desglosada de la aplicación.

Posteriormente se describen los Resultados del Trabajo Especial de Grado y, finalmente, se ofrecen las Conclusiones y Recomendaciones del mismo, así como las Referencias Bibliográficas y digitales, junto con los Anexos.

CAPÍTULO I: PLANTEAMIENTO DEL PROBLEMA

En este capítulo se describe el problema cuya solución se busca alcanzar con este trabajo de investigación, el objetivo de la investigación, los objetivos específicos, el alcance del Trabajo Especial de Grado, las preguntas de investigación y los resultados que se esperan obtener de esta investigación.

I.1 Problema de Investigación

Las personas frecuentemente dependen de recomendaciones hechas por otros en decisiones rutinarias, preguntando a amigos o colegas para seleccionar un nuevo libro, leyendo críticas antes de decidir ir a ver una película o revisando las calificaciones de compradores previos antes de adquirir un artículo en una tienda en línea (Ricci, Rokach, & Shapira, 2010). A medida que se comenzaron a desarrollar sitios web orientados al comercio electrónico, surgió la necesidad de proveer recomendaciones de forma automatizada mediante el filtrado de las alternativas disponibles. El crecimiento explosivo en la cantidad y variedad de información disponible en la *Web* y la introducción rápida de nuevos servicios de comercio electrónico (compra de productos, comparaciones, subastas, etc.) frecuentemente sobrecargó a los usuarios, llevándolos a tomar malas decisiones (Ricci, Rokach, Shapira, & Kantor, 2010). Para solventar esta necesidad surgen los sistemas de recomendación, herramientas y técnicas de software que proveen sugerencias de ítems que serán de interés para el usuario, usualmente de forma personalizada.

I.2 Justificación del Problema

La empresa ALS Telecom actualmente basa su operación en el envío de mensajes de texto de alto volumen para clientes externos. Sin embargo, la empresa también ofrece servicios de desarrollo de proyectos de información no relacionados con el envío de mensajes de texto, y trata constantemente de expandir su posición incursionando en nuevos mercados. Si bien la empresa ha participado en desarrollos *Web* para el consumo interno y externo de varios clientes, no ha estado involucrada en un desarrollo orientado al público. Con este proyecto la empresa desea expandir su oferta incursionando en el terreno de las redes sociales, por lo que ha ideado un portal para la publicación, intercambio y administración de contenido literario, asociado a frases célebres y pensamientos; pensando en poder ofrecer un ejemplo tangible de sus servicios y en monetizar el tráfico generado por el portal. La necesidad de diferenciarse de la competencia ha destacado la conveniencia de desarrollar un motor de recomendación de contenidos para dicho portal.

I.3 Objetivo General del Trabajo Especial de Grado

Desarrollar un motor de recomendación de contenido incluido en un portal web orientado al consumo, publicación e intercambio de frases célebres.

I.4 Objetivos Específicos

1. Levantar los requerimientos de la aplicación para definir sus lineamientos.
2. Definir e implementar la base de datos a ser usada por el portal web con información de entrada para el sistema de recomendaciones.
3. Desarrollar una aplicación *Web* que permita la inclusión, publicación, administración y consulta de frases literarias.
4. Desarrollar un módulo de filtrado colaborativo que permita la sugerencia de otros documentos al usuario basado en el comportamiento de sus pares.
5. Integrar el portal web a las redes sociales escogidas por el cliente.
6. Realizar las pruebas pertinentes al portal web y al sistema de recomendación.
7. Documentar la implementación de las funcionalidades del sistema.
8. Elaborar documentación para la administración y extensión de la plataforma.

I.5 Alcance

El alcance propuesto contempla el diseño y desarrollo de un portal web para permitir la interacción del usuario con el sistema, la administración del contenido literario del portal y los usuarios del mismo y la interacción con redes sociales, así como la creación del sistema de recomendación de frases para los usuarios de la aplicación.

Para lograr este alcance, se abarcaron los siguientes elementos:

- Diseño de una base de datos que permita satisfacer los requerimientos de la aplicación, tanto del portal como del sistema de recomendación.
- Diseño y desarrollo de una aplicación web para la inclusión, publicación y consulta de frases literarias.
- Integración de la aplicación web con las redes sociales, ofreciendo la posibilidad de compartir contenido en Twitter, Facebook y Google+, y de utilizar cuentas existentes de Facebook para conectarse a la aplicación y para realizar comentarios.
- Implementación de un sistema que recomiende a los usuarios frases literarias de forma individual de acuerdo a las preferencias del conjunto.

I.6 Limitaciones

Las limitaciones de la presente investigación son las siguientes:

- Se debe construir una versión funcional del portal web antes de comenzar la implementación del sistema de recomendación, para lograr obtener un conjunto de datos de prueba para el mismo. Como tal, y tomando en cuenta las limitaciones de tiempo, el conjunto de pruebas fue construido a base de un pequeño número de usuarios.
- Se debe realizar una adquisición de contenido progresiva, por lo que se anticipó realizar el despliegue de la aplicación web con un estimado de 5.000 ítems en la base de datos.
- El sistema de recomendación a construir utiliza únicamente la información recopilada de las calificaciones internas de la aplicación y la misma tendrá la forma de calificación de 0,5 a 5 en incrementos de 0,5, al ser el estándar observado para calificaciones.
- El despliegue de la aplicación web se realizó en un ambiente LAMP (Linux, Apache, MySQL, PHP) remoto, por ser el hospedaje proveído por el cliente para el mismo.

CAPÍTULO II: MARCO TEÓRICO CONCEPTUAL

En este capítulo se describen las teorías, conceptos y procesos relacionados con el problema a resolver. También se indica cual es el estado del arte, es decir, si ya existen soluciones similares se indican o describen, destacando cuáles son sus debilidades y que es lo que la aplicación a desarrollar mejora.

II.1 Sistemas de recomendación

II.1.1 Definición

Un sistema de recomendación (en adelante SR) se define como un sistema de filtrado de información que intenta recomendar ítems o elementos sociales significativos a un conjunto de usuarios. Se denota comúnmente como ítem el elemento específico que el sistema recomienda a los usuarios. Un sistema de recomendación se enfoca, normalmente, en un tipo específico de ítem y se personaliza su diseño, interfaz grafica y técnica principal de recomendación para proveer sugerencias útiles y efectivas para ese tipo específico de ítem (Ricci, Rokach, & Shapira, 2010).

El problema de recomendación puede ser definido como estimar la respuesta de un usuario para un nuevo ítem, basándose en información histórica almacenada en el sistema, y sugerir a este usuario ítems originales para los cuales la respuesta predicha es alta. El tipo de respuestas *usuario-ítem* varían de una aplicación a otra, y pueden ser agrupadas en escalares, binarias y unarias. Las respuestas escalares, también conocidas como clasificaciones, son valores numéricos (por ejemplo: 1 a 5 estrellas) u ordinales (por ejemplo: muy de acuerdo, de acuerdo, neutral, en desacuerdo, muy en desacuerdo) que representan los posibles niveles de apreciación de usuarios por ítems. Las respuestas binarias solo tienen dos niveles posibles que codifican niveles de apreciación opuestos (por ejemplo: gusto y disgusto o interesado y no interesado). Finalmente, las respuestas unarias capturan la interacción de un usuario con un ítem (por ejemplo: compra, búsqueda, lectura en línea) sin dar información explícita de la apreciación del usuario por el ítem. Dado que la mayoría de los usuarios tiende a interactuar con ítems que encuentran interesantes, las respuestas unarias aun proveen información útil acerca de las preferencias de los usuarios (Desrosiers & Karypis, 2011).

La forma en la que las respuestas del usuario pueden ser capturadas también puede diferir. Los usuarios del sistema pueden dar calificaciones explícitas después de consumir un ítem y también se pueden obtener implícitamente del historial de compras o los patrones de acceso al sistema (Desrosiers & Karypis, 2011). Para simplificar, se denominara como calificación a cualquier tipo de respuesta usuario-ítem a lo largo de este documento.

II.1.2 Antecedentes

El primer sistema de recomendación comercial, llamado Tapestry, fue diseñado en 1992 para recomendar documentos de grupos de discusión a un conjunto de usuarios, tratando de incentivar la colaboración social para que los usuarios no se viesan inundados por un gran volumen de documentos. Este tipo de sistema se denominó con el término de “filtrado colaborativo”, y busca analizar datos de uso a lo largo de los usuarios para encontrar pares usuario-ítem adecuados. El filtrado colaborativo se yuxtapuso contra la metodología de filtrado de contenido, que tuvo sus orígenes en el campo de la recuperación de información, y que no se basa explícitamente en la información de todos los usuarios. Algunos éxitos tempranos de filtrado colaborativo en dominios relacionados incluyeron el sistema GroupLens en 1994 (Melville & Sindhvani, 2010).

Las formulaciones iniciales para sistemas de recomendación estuvieron basadas en estadísticas de correlación sencillas y modelado predictivo. El problema de filtrado colaborativo se tomó por las prácticas de estadística y aprendizaje de máquina, donde se correlacionó con problemas de clasificación, lo que permitió utilizar técnicas de reducción de dimensionalidad para mejorar la calidad de las soluciones. Concurrentemente, varios esfuerzos intentaron combinar métodos basados en contenido con filtrado colaborativo e incorporar conocimiento adicional del dominio en la arquitectura de los sistemas de recomendación (Melville & Sindhvani, 2010).

La disponibilidad de conjuntos de datos en la web, junto con el interés generado por la directa relevancia al comercio electrónico, resultó en un avance en la investigación en el área. Netflix, un sistema de alquiler de videos en línea y DVDs, publicó un conjunto de 100 millones de calificaciones hechas por medio millón de usuarios a miles de películas y anunció una competencia abierta para el mejor algoritmo de filtrado colaborativo en este dominio en 2007, de donde surgieron técnicas de factorización y análisis estadístico de matrices.

Actualmente, los sistemas de recomendación son un área activa de investigación, con una conferencia dedicada de la ACM (*Association for Computing Machinery*), intersectando varias subdisciplinas de estadística, aprendizaje de máquina, minería de datos y recuperación de información. Se han aplicado sistemas de recomendación en diversos dominios que van desde recomendar páginas web a música, libros y otros productos orientados al consumidor.

II.1.3 Funcionalidades de los sistemas de recomendación

Los sistemas de recomendación son utilizados por un usuario final, pero existen varias razones por las cuales un proveedor de servicio puede estar interesado en implementar esta tecnología como parte de sus soluciones (Herlocker, Konstan, Terveen, & Riedl, 2004):

- Incrementar el número de ítems consumidos: Esta es probablemente la función más importante para un sistema de recomendación, incrementar el número de usuarios que

aceptan las recomendaciones y consumen un ítem comparado con el número de visitantes que navegan por la información.

- Colocar ítems con mayor diversidad: otra función mayor de un SR es permitir al usuario seleccionar ítems que pueden ser difíciles de encontrar sin una recomendación explícita. El costo de promocionar un ítem de popularidad media o baja puede ser alto, mientras que un SR puede sugerir dichos ítems a los usuarios adecuados.
- Incrementar la satisfacción del usuario: la combinación de recomendaciones efectivas y una interfaz usable incrementará la evaluación subjetiva del sistema por parte del usuario. Esto, a su vez, incrementará el uso del sistema y la probabilidad de que las recomendaciones sean aceptadas.
- Incrementar la fidelidad del usuario: un usuario debería ser leal a un sistema que, al ser usado, reconozca a un usuario antiguo. A medida que el usuario interactúa por más tiempo con el sistema la representación de sus preferencias será más refinada y las salidas del SR pueden ser personalizadas para ajustarse a las preferencias del usuario.
- Comprender mejor lo que el usuario quiere: un SR describe las preferencias del usuario y el proveedor del servicio puede decidir reutilizar ese conocimiento para otras metas, como mejorar el manejo del inventario o la producción de un ítem, entre otros.

Los usuarios también tienen razones para interactuar con un SR y, frecuentemente, el mismo debe balancear las necesidades de los usuarios y el proveedor para ofrecer un servicio que sea valioso para ambos. Se han documentado once tareas primarias que normalmente se asocian con un SR (Ricci et al., 2010):

- Encontrar algunos ítems adecuados: recomendar al usuario algunos ítems como una lista ordenada con predicciones de cuánto le gustaría al usuario cada ítem. Esta es la tarea primordial que la mayoría de los SR comerciales resuelve, aunque algunos no indican explícitamente la calificación predicha.
- Encontrar todos los ítems adecuados: recomendar todos los ítems que pueden satisfacer algunas necesidades del usuario. Esta tarea puede ser especialmente importante cuando el número de ítems es relativamente pequeño o cuando el SR es de importancia crítica.
- Resaltado en contexto: enfatizar algún ítem de un contexto dado dependiendo de las preferencias a largo plazo del usuario. Por ejemplo, un SR para televisión podría resaltar los programas interesantes de la guía de los mismos.
- Recomendar una secuencia: en vez de generar una única recomendación, la idea es recomendar una secuencia de ítems que son adecuados como un todo. Algunos ejemplos típicos son series de libros, compilaciones de pistas musicales o libros relacionados por algún tópico.
- Recomendar un conjunto: sugerir un grupo de ítems que concuerdan juntos. Por ejemplo, un plan de viaje puede estar compuesto de diferentes atracciones, destinos y servicios en un área delimitada, pero desde el punto de vista del usuario estas alternativas pueden ser consideradas como un único destino de viaje.
- Solo ojear: En esta tarea, el usuario navega por el catálogo de ítems sin la intención de consumir alguno, y la tarea del SR es ayudar al usuario a navegar por los ítems con

mayor probabilidad de estar dentro del alcance de los intereses del usuario por la sesión de navegación.

- Encontrar un sistema de recomendación creíble: algunos usuarios no confían en los SR y juegan con ellos para evaluar su eficacia. Por lo tanto, algunos SR ofrecen funciones específicas para permitir a los usuarios probar su comportamiento en adición a las requeridas para realizar recomendaciones.
- Mejorar el perfil: el sistema permite al usuario proveer información acerca de sus gustos. Esta es una tarea fundamental que es estrictamente necesaria para proveer recomendaciones personalizadas. Si el sistema no tiene conocimiento específico del usuario activo solo podrá proveer las mismas recomendaciones que realizaría para un usuario “promedio”.
- Permitir al usuario expresarse: a algunos usuarios no les importan las recomendaciones, sino que se les permita contribuir con sus calificaciones y expresar sus opiniones. La satisfacción por esta actividad aún puede ser un fuerte motivador para atar al usuario a la aplicación.
- Ayudar a otros: Algunos usuarios están felices de contribuir información dado que creen que la comunidad se beneficia de su contribución. Esto también puede ser un gran motivador para introducir información en un SR que no se utiliza de forma rutinaria.
- Influenciar a otros: en los SR basados en la Web existen usuarios cuya meta es explícitamente influenciar a otros a consumir ítems particulares. De hecho, existen también usuarios maliciosos que pueden utilizar el sistema únicamente para promover o penalizar ciertos ítems.

II.1.4 Datos y fuentes de conocimiento

Los SR son sistemas de procesamiento de información que activamente reúnen varios tipos de datos para construir sus recomendaciones. Los datos principalmente embarcan los ítems a sugerir y los usuarios que recibirán las recomendaciones pero, dado a la diversidad de fuentes de conocimiento disponibles para los SR, la utilidad de cada fuente de datos dependerá finalmente de la técnica de recomendación escogida. En general, hay técnicas que utilizan datos muy simples, como calificaciones o evaluaciones de los ítems y técnicas dependientes de conocimiento detallado del dominio, como descripciones ontológicas de los usuarios u ítems, restricciones de dominio o de las relaciones sociales y actividades de los usuarios. Como una clasificación general, los datos usados por un SR se refieren a tres tipos de objetos: ítems, usuarios y transacciones, o sea, las relaciones entre los usuarios y los ítems (Ricci et al., 2010).

II.1.4.1 Ítems

Los ítems son los objetos a ser recomendados y pueden ser caracterizados por su complejidad y su valor o utilidad. El valor de un ítem es positivo si el ítem es útil para el usuario y negativo si el ítem no es apropiado para el mismo, teniendo en cuenta que cuando un usuario consume un ítem siempre incurre en un costo, que incluye el costo cognitivo y temporal de buscar y consumir el ítem así como el costo monetario que pueda o no pagar por el mismo.

Algunos ítems con baja complejidad y valor son noticias, páginas web, libros, CDs o películas. Ítems con mayor complejidad son cámaras digitales, teléfonos móviles, etc. Algunos de los ítems más complejos son pólizas de seguro, inversiones financieras y viajes, entre otros.

Los SR, de acuerdo a su tecnología principal, pueden usar un conjunto de propiedades y características de los ítems, para describirlos o comprender como las propiedades afectan la utilidad de un ítem. Los ítems pueden ser representados utilizando varios enfoques de información, de forma minimalista, como un único identificador, o en una forma más compleja, como un conjunto de atributos o un concepto en una representación ontológica del dominio (Ricci et al., 2010).

II.1.4.2 Usuarios

Un SR explota un conjunto de información de los usuarios para personalizar las recomendaciones generadas. Esta información puede ser estructurada de varias formas dependiendo de la técnica de recomendación utilizada. El modelo del usuario codifica sus preferencias y necesidades dentro de un perfil. El modelo del usuario siempre será central, a menos que las recomendaciones no sean personalizadas, ya que éste conforma parte central en la personalización de las mismas.

Los usuarios también pueden ser descritos por datos de su patrón de comportamiento, como por los patrones de navegación en sistemas web o patrones de búsqueda de viaje en un SR de viajes. Adicionalmente, los datos del usuario pueden incluir relaciones entre usuarios y el SR puede utilizar esta información para recomendar ítems a usuarios que fueron preferidos por usuarios similares o usuarios relacionados (Ricci et al., 2010).

II.1.4.3 Transacciones

Se definen transacciones, de forma general, como interacciones almacenadas entre un usuario y el SR. Las transacciones contienen información importante generada durante la interacción del usuario con el sistema, útil para el algoritmo de generación de recomendaciones que utiliza el sistema. Para dar un ejemplo, un registro de transacciones puede contener una referencia al ítem seleccionado por el usuario y una descripción del contexto de esa recomendación particular, así como reacciones explícitas que el usuario pueda proveer.

En líneas generales, la calificación es la forma más popular de transacción que almacena un SR, generalmente de forma explícita. Otra forma de evaluación del usuario consiste en etiquetas asociadas por el usuario con el ítem presentado por el sistema. En transacciones que almacenan calificaciones implícitas el sistema intenta inferir la opinión del usuario basado en sus acciones, por ejemplo, al seleccionar un ítem como producto de una búsqueda.

II.1.5 Paradigmas de recomendación

En esta sección se presentan diferentes tipos de sistemas de recomendación, que varían en términos del dominio al que están orientados, el conocimiento usado y, especialmente, en el algoritmo de recomendación. La siguiente taxonomía se ha convertido en la forma clásica de distinguir entre los diferentes SR y distingue entre 6 diferentes enfoques de recomendación: basado en contenido, filtrado colaborativo, demográfico, basado en conocimiento, basado en comunidad y sistemas híbridos (Ricci et al., 2010).

II.1.5.1 Sistemas de recomendación basados en contenido

El sistema aprende a recomendar ítems que son similares a los que le han gustado al usuario en el pasado. Estos SR analizan un conjunto de documentos y/o descripciones de ítems previamente calificadas por un usuario y construyen un modelo o perfil de intereses de usuario basado en las características de los objetos calificados por dicho usuario. El perfil es una representación estructurada de intereses del usuario, adaptados para recomendar nuevos ítems interesantes. El proceso de recomendación consiste, básicamente, en cotejar los atributos del perfil del usuario contra los atributos de un ítem, tomando como resultado una estimación de relevancia que representa el nivel de interés del usuario en el ítem (Lops, de Gemmis, & Semeraro, 2010).

En general, estos SR tienen la ventaja de poder recomendar ítems no calificados a usuarios con intereses únicos y proveer explicaciones a las recomendaciones y, si los ítems tienen muchos atributos, una técnica de filtrado por contenidos puede ser muy precisa. Sin embargo, los métodos basados en contenido requieren una gran cantidad de información por parte del usuario y no tienen un gran desempeño con contenido que no tiene muchos atributos. Adicionalmente, estas técnicas requieren analizar el contenido y la categorización de los ítems, que puede no ser posible o incurrir en un gran costo computacional (Haw & Watermilller, 2010).

II.1.5.2 Sistemas de recomendación basados en filtrado colaborativo

Los enfoques basados en filtrado colaborativo analizan las calificaciones de todos los usuarios del sistema, teniendo como principio que es probable que la calificación de un usuario X para un nuevo ítem I sea similar a la de otro usuario Y, si X y Y han calificado otros ítems de forma similar. De igual forma, es probable que X califique dos ítems I y J de forma similar, si otros usuarios han dado calificaciones similares a estos dos ítems. Los métodos de filtrado colaborativo se pueden subdividir en dos clases generales: basados en vecindario o basados en un modelo (Desrosiers & Karypis, 2011).

En el filtrado colaborativo basado en *vecindario* las calificaciones usuario-ítem almacenadas en el sistema son usadas directamente para predecir calificaciones para nuevos ítems, y esto puede ser hecho enfocándose en los ítems o en los usuarios. Los sistemas basados en usuarios evalúan el interés de un usuario X en un ítem I utilizando los ratings para este ítem

de otros usuarios llamados *vecinos*, que tienen patrones de calificación similares. Los enfoques basados en ítems, por otro lado, predicen la calificación de un usuario X para un ítem I basado en las calificaciones de X para ítems similares a I (Desrosiers & Karypis, 2011).

En contraste con los métodos basados en vecindario, los métodos basados en un modelo utilizan las calificaciones para aprender un modelo predictivo. La idea general es modelar las interacciones usuario-ítem con factores que representan las características latentes de los usuarios e ítems del sistema. El modelo es luego entrenado con los datos disponibles y finalmente utilizado para predecir las calificaciones de los usuarios a los ítems (Desrosiers & Karypis, 2011).

El filtrado colaborativo tiene como ventajas poder funcionar en dominios donde se involucran atributos subjetivos o donde no existe mucho contenido asociado con un ítem. Adicionalmente, es más sencillo para un SR basado en filtrado colaborativo producir recomendaciones novedosas que parecen no tener relación con el perfil del usuario. Sin embargo, tienen problemas graves cuando existen errores en los datos capturados de las calificaciones, así como en los inicios en frío cuando existen muy pocas calificaciones. Adicionalmente, los datos almacenados por el sistema son frecuentemente dispersos dada la disparidad entre el número de usuarios y el número de ítems en un sistema común (Haw & Watermillar, 2010).

II.1.5.2.1 Recomendaciones basadas en ítems versus basadas en usuarios

Al momento de escoger entre la implementación de un SR de vecindario basado en ítems o en usuarios, se pueden considerar los siguientes cinco criterios (Desrosiers & Karypis, 2011):

- **Precisión:** la precisión de los métodos de vecindario depende principalmente de la tasa entre el número de usuarios y el número de ítems en el sistema. En general, un pequeño número de vecinos de alta confianza es preferible a un gran número de vecinos para los que las métricas de similitud no son confiables. Por lo tanto, en casos donde el número de usuarios es mucho mayor que el número de ítems los métodos basados en ítems pueden producir recomendaciones más precisas y el caso contrario puede beneficiarse más de utilizar métodos basados en usuarios.
- **Eficiencia:** la eficiencia computacional y de memoria de un SR también depende de la tasa entre el número de usuarios e ítems, de tal forma que, cuando los usuarios exceden a los ítems, los enfoques basados en ítems requieren mucha menos memoria y tiempo para computar los índices de similitud que los enfoques basados en usuarios, haciéndolos más escalables.
- **Estabilidad:** la frecuencia y cantidad de cambios en los usuarios e ítems del sistema también influye en la elección. Si la lista de ítems disponibles es razonablemente estática en comparación a los usuarios un método basado en ítems podrá ser preferible ya que los índices de similitud de los ítems podrán ser calculados en intervalos de tiempo de

baja frecuencia manteniendo la operatividad del SR. En el caso contrario, los métodos basados en usuarios han probado ser más estables.

- Justificación: los métodos basados en ítems pueden ser utilizados para justificar una recomendación fácilmente, utilizando la lista de los ítems vecinos y los índices de similitud para presentar al usuario la explicación, siendo posible permitir al usuario modificar la lista de vecinos y sus índices para participar directamente en el proceso de recomendación. Los métodos basados en usuario, sin embargo, tienen más dificultades para entregar explicaciones porque el usuario no conoce a los otros usuarios que actúan de vecinos en la recomendación.
- Descubrimiento fortuito: los SR que utilizan métodos basados en ítems tienden a recomendar al usuario ítems que están relacionados a los que el usuario usualmente aprecia, lo que puede conllevar a recomendaciones seguras pero que aportan menos ayuda al usuario para descubrir diferentes tipos de ítems que le puedan gustar en la misma medida. Los enfoques basados en usuarios, por otro lado, tienen mayor probabilidad de entregar recomendaciones fortuitas, particularmente si la recomendación se hace con un número pequeño de vecinos cercanos.

II.1.5.2.2 Cálculo de índices de similitud

Los índices de similitud juegan un rol doble en los métodos basados en vecindario: permiten la selección de vecinos confiables, cuyas calificaciones son usadas en la predicción, y proveen los medios para dar mayor o menor importancia a estos vecinos en la misma. El cálculo de los índices de similitud es uno de los aspectos más críticos al construir un SR basado en vecindario, ya que puede tener un impacto significativo en su precisión y desempeño.

II.1.5.2.2.1 Similitud basada en correlación

Una medida de la similitud entre dos ítems i y j utilizada comúnmente en los SR es el coeficiente de correlación de Pearson (PC), el cual se presenta en la fórmula de la figura 1:

$$PC(i, j) = \frac{\sum_{u \in \mathcal{U}_{ij}} (r_{ui} - \bar{r}_i)(r_{uj} - \bar{r}_j)}{\sqrt{\sum_{u \in \mathcal{U}_{ij}} (r_{ui} - \bar{r}_i)^2 \sum_{u \in \mathcal{U}_{ij}} (r_{uj} - \bar{r}_j)^2}}.$$

Figura 1: Ecuación para calcular el coeficiente de correlación de Pearson entre dos ítems (Desrosiers & Karypis, 2011)

Las diferencias en las escalas de calificación de los usuarios individuales son usualmente más pronunciadas que las diferencias en las calificaciones dadas a ítems individuales, por lo que, al calcular las similitudes entre ítems, puede ser más apropiado comparar calificaciones centradas en la media del usuario en vez de la del ítem. La similitud por el coseno ajustado (AC, ver figura 2) es una modificación de la correlación de Pearson que compara las calificaciones centradas en la media del usuario y, en algunos casos, tiene mejor desempeño que la correlación de Pearson para predecir calificaciones utilizando métodos basados en ítems (Desrosiers & Karypis, 2011).

$$AC(i, j) = \frac{\sum_{u \in \mathcal{U}_{ij}} (r_{ui} - \bar{r}_u)(r_{uj} - \bar{r}_u)}{\sqrt{\sum_{u \in \mathcal{U}_{ij}} (r_{ui} - \bar{r}_u)^2 \sum_{u \in \mathcal{U}_{ij}} (r_{uj} - \bar{r}_u)^2}}.$$

Figura 2: Ecuación para calcular la similitud por coseno ajustado entre dos ítems (Desrosiers & Karypis, 2011)

Finalmente, la tercera medida de similitud común en los SR basados en vecindad es el rango de correlación de Spearman (SRC, ver figura 3). Mientras que la correlación de Pearson utiliza los valores de las calificaciones directamente el SRC considera la clasificación de estas calificaciones. Denotando k_{ui} el rango de calificación del ítem i en la lista de ítems calificados del usuario u , la similitud entre dos ítems u y v mediante el SRC se evalúa como:

$$\text{SRC}(u, v) = \frac{\sum_{i \in \mathcal{I}_{uv}} (k_{ui} - \bar{k}_u)(k_{vi} - \bar{k}_v)}{\sqrt{\sum_{i \in \mathcal{I}_{uv}} (k_{ui} - \bar{k}_u)^2 \sum_{i \in \mathcal{I}_{uv}} (k_{vi} - \bar{k}_v)^2}},$$

Figura 3: Ecuación para calcular la similitud por rango de correlación de Spearman entre dos ítems (Desrosiers & Karypis, 2011).

II.1.5.2.2 Selección del vecindario

El número de vecinos más cercanos a seleccionar y el criterio utilizado para realizar la selección pueden, a su vez, tener un serio impacto en la calidad del SR. La selección de los vecinos usados en la recomendación de ítems se hace, típicamente, en dos pasos: primero se filtra globalmente, manteniendo solo los candidatos más probables y, posteriormente, un paso por predicción en el que se escoge los mejores candidatos para la predicción actual.

El pre filtrado de vecinos es un paso esencial que hace los enfoques basados en vecindad prácticos reduciendo la cantidad de índices de similitud a almacenar y limitando el número de vecinos candidatos para considerar en las predicciones. En general, el filtrado se realiza de una o más de las siguientes formas (Desrosiers & Karypis, 2011):

- Filtrado de los N- tope: para cada usuario o ítem solo se mantiene una lista de los N vecinos más cercanos y sus respectivas similitudes. Si N es muy grande, el SR requiere de una cantidad excesiva de memoria para almacenar las listas de la vecindad y las predicciones de calificaciones se enlentecen. Por otra parte, si se selecciona un valor de N muy pequeño, se puede reducir la cobertura del método de recomendación, lo que causa que algunos ítems no sean recomendados nunca.
- Umbral de filtrado: en lugar de mantener un número fijo de vecinos cercanos, este enfoque mantiene todos los vecinos cuyas calificaciones de similitud tengan una magnitud mayor a un umbral dado. Si bien este enfoque es más flexible que el filtrado de N tope, el valor correcto del umbral puede ser difícil de determinar.
- Filtrado negativo: en general, las correlaciones de calificaciones negativas son menos confiables que las positivas. Aunque las correlaciones negativas pueden indicar membrecía a diferentes grupos, no indican cuan diferentes son esos grupos o si son compatibles para otras categorías o ítems. Sin embargo, que las correlaciones negativas puedan o no ser descartadas depende enteramente de los datos.

Una vez que se computa la lista de vecinos candidatos la predicción de nuevas calificaciones se realiza, normalmente, con los k vecinos más cercanos, es decir, los k vecinos cuyo índice de similitud tiene la mayor magnitud. El problema importante es escoger el valor de k . La precisión de las predicciones observada para valores incrementales de k típicamente sigue

una función cóncava. Por lo tanto, cuando el número de vecinos se restringe utilizando un k pequeño, por ejemplo $k < 20$, la precisión de la predicción es normalmente baja. A medida que k incrementa, más vecinos contribuyen a la predicción y la varianza introducida por los vecinos individuales se promedia y, como resultado, la precisión de las predicciones aumenta. Finalmente, cuando se utilizan demasiados vecinos en la predicción, por ejemplo $k > 50$, la precisión cae ya que las pocas relaciones locales fuertes se diluyen por la gran cantidad de relaciones débiles. Aunque un número de vecinos entre 20 y 50 se describe con mayor frecuencia en la literatura, el valor óptimo de k debería ser determinado por validación cruzada (Desrosiers & Karypis, 2011).

II.1.5.3 Sistemas de recomendación demográficos

Este tipo de SR recomienda ítems basados en el perfil demográfico del usuario. El SR asume que se deberían generar recomendaciones diferentes para diferentes nichos demográficos. Muchos sitios web adoptan opciones de personalización simples y efectivas basadas en demográficas. Por ejemplo, los usuarios son despachados a sitios web particulares dependiendo de su idioma o país, o se realizan sugerencias dependiendo de la edad del usuario. Si bien estos sistemas han tenido popularidad en literatura de mercadeo, ha habido poca investigación en el área de SR demográficos (Ricci et al., 2010).

II.1.5.4 Sistemas de recomendación basados en conocimiento

Los SR basados en conocimiento recomiendan ítems basados en conocimiento específico del dominio del sistema acerca de cómo ciertas características de los ítems concuerdan con las necesidades y preferencias de los usuarios y, finalmente, que tan útil es el ítem para los mismos. Dos tipos de SR basados en conocimiento se basan en casos de uso y en restricciones. Los SR basados en casos de uso utilizan una función de similitud para estimar cuantas de las necesidades del usuario concuerdan con las recomendaciones, de tal forma que la similitud puede ser directamente interpretada como la utilidad del ítem para el usuario. Los sistemas basados en restricciones predominantemente explotan bases de conocimiento predefinidas que contienen reglas explícitas acerca de la relación entre los requerimientos de los usuarios y las características de los ítems.

Los SR basados en conocimiento tienden a funcionar mejor que otros al principio del desarrollo y a ser sobrepasados por otros métodos que pueden aprovechar el registro de la interacción entre el usuario y el sistema.

II.1.5.5 Sistemas de recomendación basados en comunidad

Este tipo de sistema recomienda ítems basados en las preferencias de los amigos de los usuarios. Existe evidencia que sugiere que los usuarios tienden a confiar más en las recomendaciones de sus amigos que en recomendaciones de individuos similares pero anónimos (Ricci et al., 2010). Este tipo de SR modela y adquiere información acerca de las

relaciones sociales del usuario y las preferencias de los amigos del mismo y basan la recomendación en calificaciones realizadas por dichas amistades. Estos SR son producto de la popularidad de las redes sociales, permitiendo una adquisición de datos asociados a las relaciones sociales de los usuarios de forma simple y exhaustiva (Ricci et al., 2010).

La investigación en esta área esta en fases tempranas y hasta ahora ha arrojado resultados mixtos. Algunos investigadores reportan que las recomendaciones basadas en redes sociales no son más precisas que las basadas en filtrado colaborativo, excepto en casos especiales (Massa & Avesani, 2004), mientras que otros han mostrado que en algunos casos los datos de redes sociales proveen mejores recomendaciones que datos de similitud de perfil y que agregar estos datos a enfoques de filtrado colaborativo tradicionales mejora los resultados del motor de recomendaciones (Groh & Ehmig, 2007).

II.1.6 Evaluación de Sistemas de Recomendación

En general, la evaluación de SR se realiza en tres modalidades: *offline*, en la cual se realiza un experimento con un conjunto de datos de usuarios calificando ítems, recogido previamente y, utilizándolo, se trata de simular el comportamiento de los usuarios que interactúan con el SR. Como segunda modalidad se observan los estudios de usuarios, en los cuales se recluta un conjunto de usuarios de prueba y se les solicita que realicen ciertas tareas que requieren interactuar con el SR, observando y almacenando su comportamiento simultáneamente. Finalmente, las pruebas en línea consisten en realizar una prueba en caliente, en la cual, típicamente, se redirige un pequeño porcentaje del tráfico del sistema a un motor de recomendaciones alternativo, almacenando las interacciones de los usuarios con los diferentes SR (Shani & Gunawardana, 2010).

Inicialmente, la mayoría de los SR han sido evaluados en su poder de predicción, su habilidad para predecir correctamente el comportamiento del usuario. Sin embargo, actualmente se sostiene que la precisión de las predicciones es crucial pero insuficiente para implementar un buen motor de recomendación. Dado que un SR es usado para diversas razones, se debe identificar el conjunto de propiedades que pueden influenciar el éxito de un SR en el contexto de su aplicación específica y luego evaluar cómo se comporta el sistema con respecto a estas propiedades relevantes. Entre las mayores propiedades a ser evaluadas están (Shani & Gunawardana, 2010) :

II.1.6.1 Precisión de las predicciones

La precisión de las predicciones es la propiedad mas discutida en la literatura de sistemas de recomendación. La precisión es, típicamente, independiente de la interfaz del sistema, por lo que puede ser evaluada en una prueba *offline*. En líneas generales se pueden realizar mediciones de la precisión de las predicciones de calificaciones, de las predicciones de uso o de las clasificaciones de ítems.

II.1.6.1.1 Precisión de predicciones de calificación

La métrica más popular para medir la precisión de las calificaciones predichas por el sistema es la media cuadrática del error (MCE). El sistema genera calificaciones predichas $\hat{r}_{ui}$ para un conjunto de prueba τ de pares ítem-usuario (u, i) para los cuales se conocen las calificaciones verdaderas r_{ui} . Se calcula la media cuadrática del error entre las calificaciones reales y las predichas de acuerdo con la fórmula de la figura 4.

$$MCE = \sqrt{\frac{\sum_{(u,i) \in \tau} (\hat{r}_{ui} - r_{ui})^2}{|\tau|}}$$

Figura 4: Ecuación para calcular la media cuadrática del error (Shani & Gunawardana, 2010).

Otra métrica popular utilizada actualmente para medir la precisión de las calificaciones es la media absoluta del error (MAE), observada en la figura 5. Si un conjunto de prueba contiene una distribución desbalanceada de ítems el MCE o MAE obtenido del mismo puede estar altamente influenciado por el error de pocos ítems de alta frecuencia. De necesitarse una medida del error de predicción de cualquier ítem, es preferible calcular el MCE o MAE separadamente para cada ítem y luego tomar el promedio a lo largo de todos los ítems. Un caso similar se puede aplicar para estudiar el error de predicción que podrá encontrar un usuario aleatorio.

$$MAE = \sqrt{\frac{\sum_{(u,i) \in \tau} |\hat{r}_{ui} - r_{ui}|}{|\tau|}}$$

Figura 5: Ecuación para calcular la media absoluta del error (Shani & Gunawardana, 2010).

II.1.6.1.2 Precisión de predicciones de uso

En muchas aplicaciones el SR no trata de predecir calificaciones de usuarios a ítems, sino que trata de recomendar ítems que el usuario consumirá. En una evaluación *offline* de predicciones de uso se tiene, típicamente, un conjunto de datos consistente de los ítems que ha consumido cada usuario. Posteriormente, se procede a escoger un usuario de prueba, esconder algunas de sus selecciones y pedir al SR que prediga un conjunto de ítems que consumirá el mismo. Esto produce cuatro posibles resultados para los ítems recomendados y escondidos, mostrados en la figura 6.

	Recomendado	No Recomendado
Consumido	Verdadero-Positivo (vp)	Falso-Negativo (fn)
No Consumido	Falso-Positivo (fp)	Verdadero-Negativo (vn)

Figura 6: Clasificación de los posibles resultados de la recomendación de un ítem a un usuario

Se pueden contar el número de ejemplos que caen en cada celda de la tabla y calcular las métricas de la figura 7, observando que, típicamente, para aumentar una métrica se debe disminuir otra – permitir listas de recomendación más largas frecuentemente mejora el *recall* y disminuye la precisión.

$$\textbf{Precisión} = \frac{\#vp}{\#vp + \#fp}$$

$$\textbf{Recall}(\textit{Tasa de verdaderos positivos}) = \frac{\#vp}{\#vp + \#fn}$$

$$\textbf{Tasa de falsos positivos} = \frac{\#fp}{\#fp + \#vn}$$

Figura 7: fórmulas de métricas de precisión de predicciones de uso

II.1.6.1.3 Precisión de clasificaciones de ítems

En algunas aplicaciones el interés del SR es generar listas ordenadas de ítems de acuerdo a las preferencias del usuario, tarea generalmente conocida como clasificación de ítems. Existen dos enfoques para medir la precisión de dicha clasificación: se puede determinar el orden correcto de un conjunto de ítems para cada usuario y medir que tan cerca llega el sistema de este orden correcto, o se puede intentar medir la utilidad de la calificación del sistema para el usuario.

II.1.6.2 Cobertura

En muchos casos, especialmente en filtrados colaborativos, la precisión de las predicciones aumenta con la cantidad de datos. Sin embargo, algunos algoritmos pueden proveer recomendaciones de alta calidad, pero para una pequeña porción de los ítems en los que tienen gran cantidad de información. El término cobertura puede referirse a una de las siguientes propiedades (Shani & Gunawardana, 2010).

II.1.6.2.1 Cobertura del catálogo

El término cobertura se utiliza con mayor frecuencia para referirse a la proporción de ítems que pueden ser recomendados por el SR, también conocido como cobertura del catálogo. La métrica más sencilla de cobertura del catálogo es el porcentaje de todos los ítems que pueden ser recomendados, o que son recomendados durante una prueba en caliente.

Otra métrica de cobertura del catálogo es la diversidad de ventas, que mide cómo los diferentes ítems son escogidos por los usuarios en un SR particular. Si cada ítem i cuenta con una proporción $p(i)$ de escogencias del usuario, el Índice Gini (G) es calculado utilizando la fórmula de la figura 8 (Shani & Gunawardana, 2010):

$$G = \frac{1}{n-1} \sum_{j=1}^n (2j - n - 1)p(i_j)$$

Figura 8: Ecuación para el cálculo del índice Gini (Shani & Gunawardana, 2010) .

Donde $i_1, \dots, i_n$ es la lista de ítems ordenados de acuerdo a $p(i)$. El índice es 0 cuando todos los ítems se escogen con igual frecuencia y 1 cuando un ítem es escogido en todos los casos.

II.1.6.2.2 Cobertura del espacio de usuarios

La cobertura también puede ser la proporción de usuarios o interacciones de usuarios para los que el sistema puede recomendar ítems. En varias aplicaciones el SR puede no proveer recomendaciones para algunos usuarios (por ejemplo, si la confianza en las predicciones para dicho usuario es muy baja) y en dichos casos se preferirá un SR que pueda proveer recomendaciones para un mayor rango de usuarios, teniendo en cuenta el intercambio entre cobertura y precisión.

La cobertura del espacio de usuarios puede ser medida por la riqueza del perfil del usuario requerido para hacer una recomendación. Por ejemplo, en el caso de filtrado colaborativo, la cobertura se puede medir como el número de ítems que el usuario debe calificar antes de recibir una recomendación que cumpla con ciertos parámetros.

II.1.6.2.3 Arranque en frío

Finalmente, se puede evaluar el desempeño del sistema con respecto a nuevos ítems y nuevos usuarios. El arranque en frío puede ser considerado un subproblema de cobertura porque mide la cobertura del sistema sobre un conjunto específico de ítems y usuarios,

buscando determinar el tamaño de dicho conjunto así como la precisión del sistema para el mismo (Shani & Gunawardana, 2010).

Al enfocarse en ítems nuevos, se debe establecer un umbral para determinar el conjunto de ítems en arranque frío, tomando como ítems fríos aquellos que no tengan calificaciones o evidencia de uso, o ítems que existan en el sistema por una cantidad de tiempo menor a un umbral establecido. De esta forma, se puede alterar un SR para que haga mejores recomendaciones para ítems fríos, generalmente a costa de una reducción en la precisión para ítems más antiguos, característica deseable para algunos sistemas.

Los métodos para el aprendizaje de las preferencias del usuario se clasifican en implícitos, que observan el comportamiento de los usuarios y realizan inferencias basadas en el mismo, y explícitos, que recogen conocimiento de hechos específicos introducidos directamente por el usuario en el sistema. El método explícito más común para un usuario nuevo consiste en presentarle una lista de ítems del sistema y solicitar que los califique. La selección de los ítems puede realizarse aleatoriamente, por popularidad, o mediante la entropía de cada ítem, que representa la dispersión de las opiniones de los usuarios sobre el mismo, dada por la fórmula de la figura 9, donde p_i la fracción de las calificaciones del ítem a_t igual a i (Rashid, Karypis, & Riedl, 2008):

$$H(a_t) = - \sum p_i \log_2(p_i)$$

Figura 9: Fórmula para calcular la entropía de un ítem (Rashid, Karypis, & Riedl, 2008).

II.1.6.3 Confianza

Así como la precisión de las predicciones de un SR tiende a mejorar a medida que crecen los datos, la confianza en las predicciones realizadas también tiende a aumentar. En varios casos el usuario puede beneficiarse al observar la confianza específica en una predicción, por ejemplo decidiendo investigar un ítem que fue recomendado con un valor de confianza bajo (Shani & Gunawardana, 2010).

La métrica de confianza más común es la probabilidad de que el valor predicho sea correcto, o el intervalo de confianza alrededor del valor predicho donde se encuentra una porción predefinida de los valores correctos, como el 95%. Dados dos SR que se comportan similarmente en precisión, puede ser deseable escoger el SR que pueda proveer estimados de confianza válidos.

Otra medición se puede realizar al filtrar ítems recomendados donde la confianza es menor a un umbral, comparando la precisión de dos SR luego de remover los ítems de confianza baja. En estas pruebas se puede calcular una curva estimando la precisión de las predicciones para cada porción de los ítems filtrados o para diferentes umbrales, logrando así evaluar

algoritmos que no utilicen el mismo método para calcular la confianza de una predicción (Shani & Gunawardana, 2010).

II.1.6.4 Novedad

Se denominan recomendaciones novedosas aquellas que recomiendan al usuario un ítem del cual no tenía conocimiento previo. Para entender la novedad del sistema a través de experimentos offline se puede dividir el conjunto de datos en dos subconjuntos tomando como pivote el tiempo, por ejemplo, ocultando las calificaciones de usuario que ocurrieron antes o después de un momento dado, evaluando la cantidad de recomendaciones realizadas por el SR que pertenecen al subconjunto posterior al momento escogido.

Otro método para evaluar recomendaciones novedosas asume que es menos probable que un ítem popular sea novedoso, por lo que la novedad se puede tomar en cuenta utilizando una métrica de precisión que asigne mayor peso a las recomendaciones correctas de ítems no populares. Finalmente, se puede evaluar la cantidad de información nueva en una recomendación junto con la relevancia del ítem recomendado, por ejemplo, multiplicando la calificación ocultada por alguna medición de la información del ítem recomendado para producir un puntaje de novedad (Shani & Gunawardana, 2010).

II.1.6.5 Diversidad

La diversidad es comúnmente definida como el opuesto de la similitud. En algunos casos, sugerir un conjunto de ítems similares puede no ser tan útil para el usuario, porque conlleva a un mayor tiempo para explorar el rango de ítems. El método más explorado para medir diversidad utiliza similitud entre ítems, definiendo la distancia entre dos ítems utilizando sus metadatos. Posteriormente, se medirá la diversidad de una lista basada en la suma de distancias o en la distancia promedio, mínima o máxima.

La métrica de similitud entre ítems puede ser diferente en la evaluación que durante la ejecución normal del SR, siendo normal escoger métricas mas exactas a la hora de evaluar y métricas aproximadas pero de mayor velocidad para cálculos online. A medida que la diversidad aumente a costa de otras propiedades, como precisión, se pueden calcular curvas para evaluar el decrecimiento de precisión versus el aumento en diversidad o viceversa (Shani & Gunawardana, 2010).

II.1.6.6 Utilidad

Se define la utilidad de una recomendación como el valor que el sistema o el usuario gana por la misma. La utilidad se puede medir limpiamente desde la perspectiva del SR o del dueño del mismo, siendo mucho más complejo capturar y modelar la utilidad para el usuario. En una aplicación donde los usuarios califican ítems los mismos se pueden utilizar como una

métrica de utilidad, asumiendo que los ítems recomendados tienen mayor utilidad a medida que la calificación del usuario del ítem aumenta.

Mientras que, típicamente, se asignan únicamente valores positivos a las recomendaciones exitosas, se pueden asignar valores negativos a las recomendaciones no exitosas. Adicionalmente, se puede agregar un costo a cada recomendación y substraerlo de la utilidad del ítem. En cualquier caso, para cualquier función de utilidad, la evaluación estándar del SR es computar la utilidad esperada de una recomendación (Shani & Gunawardana, 2010).

II.1.7 Lineamientos para la elección de técnicas específicas

Dados los diferentes enfoques que puede tomar un SR, la elección inicial de técnicas y algoritmos a utilizar es de vital importancia. Esta sección busca mostrar lineamientos para elegir una técnica de recomendación de acuerdo al problema de recomendación específico.

II.1.7.1 Fuentes de conocimiento

Una elección fundamental para el implementador de un SR es la fuente y el tipo de conocimiento que el sistema usará. En el caso de los SR las entidades primarias que deben tenerse en cuenta son los usuarios y los ítems. Para una instancia individual de recomendación se puede dividir el conocimiento de los usuarios entre lo que se conoce del usuario objetivo y lo que se conoce de la comunidad de usuarios en general, con lo cual existen tres categorías amplias de conocimiento a tomar en cuenta (Burke & Ramezani, 2010):

- Social: conocimiento obtenido de todos los perfiles de usuario almacenados en el sistema.
- Individual: conocimiento del usuario objetivo, que es el foco principal de cada interacción específica con el SR.
- Contenido: conocimiento de los ítems que son recomendados y de su uso.

Burke y Ramezani (2010) presentan la taxonomía de fuentes de conocimiento en sistemas de recomendación reflejada en la figura 10:

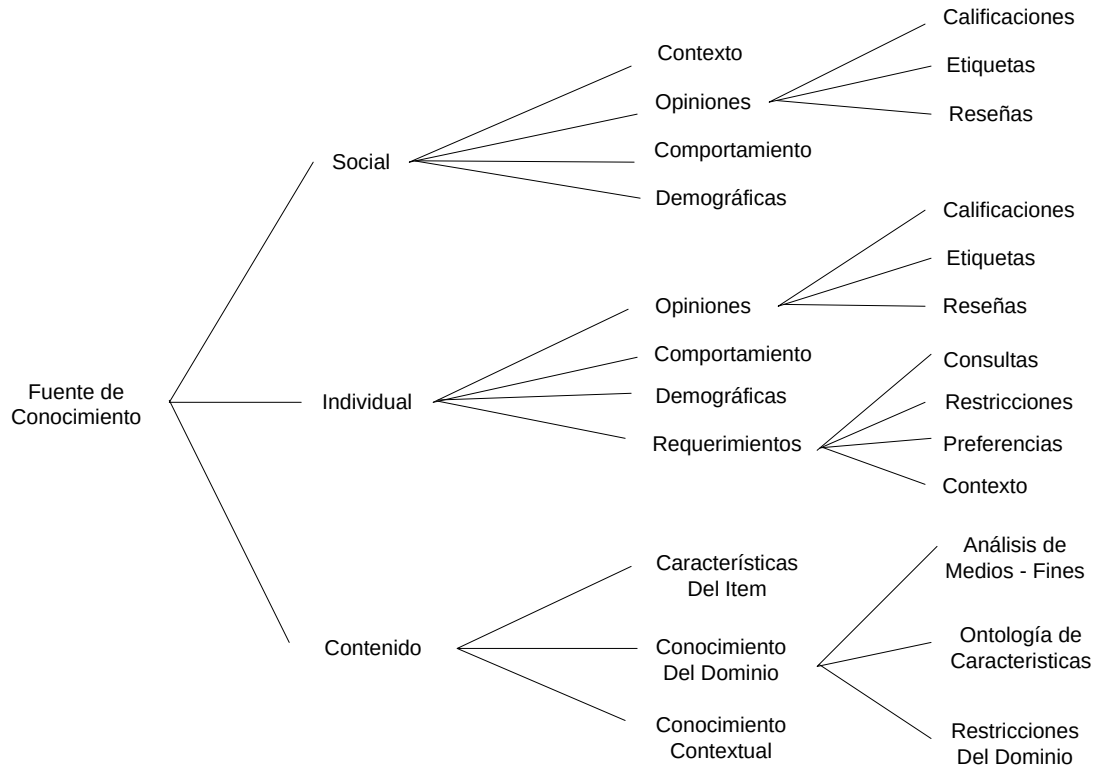


Figura 10: Taxonomía de fuentes de conocimiento en recomendación (Burke & Ramezani, 2010)

Teniendo en cuenta esta taxonomía, se pueden enumerar los tres tipos de recomendación más generales junto con las fuentes de conocimiento que utilizan, representados de forma gráfica en la figura 11 (Burke & Ramezani, 2010):

- Recomendaciones colaborativas: tratan de cotejar información de una fuente individual con una fuente de conocimiento social del mismo tipo para extrapolar las preferencias del usuario seleccionado en base a la de sus pares. Sin embargo, los requerimientos individuales son utilizados únicamente como filtros, o totalmente ignorados.
- Recomendaciones basadas en contenido: éstas se enfocan en las fuentes individuales, utilizando las características de los ítems y las opiniones del usuario para entrenar un clasificador que puede predecir las preferencias del usuario para nuevos ítems.
- Recomendaciones basadas en conocimiento: estas últimas utilizan cualquier tipo de conocimiento del dominio más sustantivo que las características de los ítems.

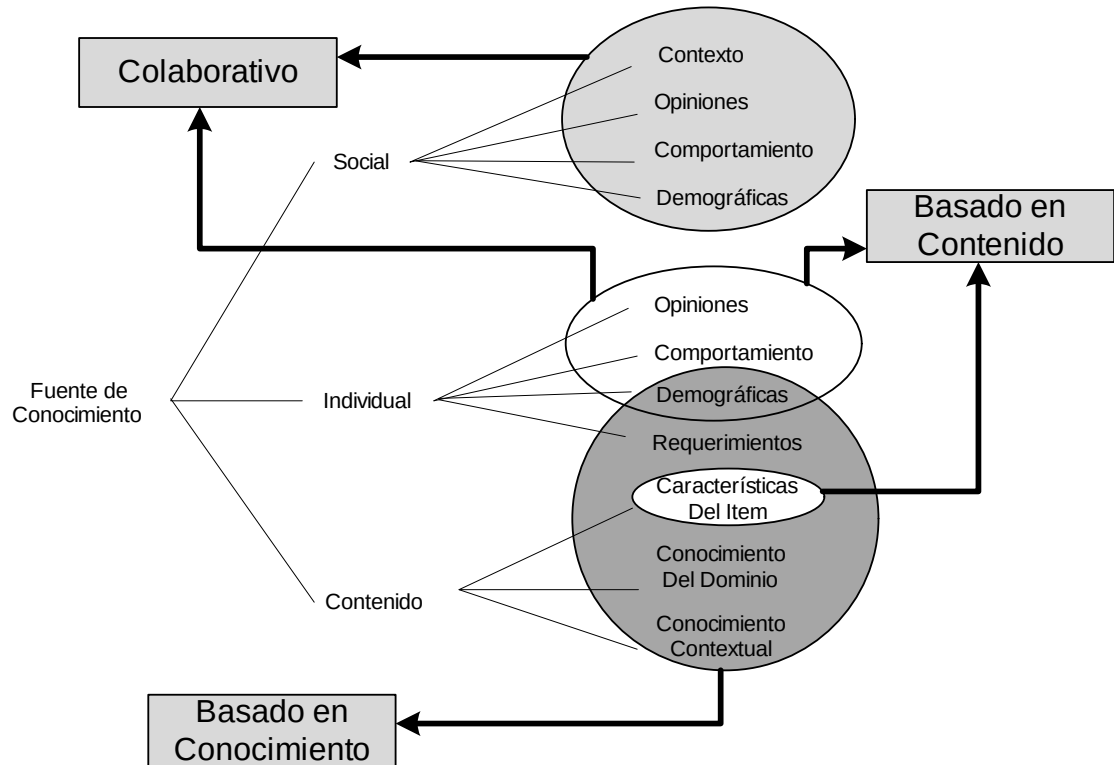


Figura 11: Fuentes de conocimiento y tipos de recomendación (Burke & Ramezani, 2010).

II.1.7.2 Dominio

El dominio de recomendación es el conjunto de ítems sobre los que opera un SR, pero también incluye el conjunto de metas o propósitos para los que se diseña el mismo. Burke y Ramezani (2010) identifican las siguientes seis características del dominio a ser consideradas para la implementación de un SR: heterogeneidad, riesgo, rotación, estilo de interacción, estabilidad de preferencias y capacidad de explicación.

- **Heterogeneidad:** Un espacio de ítems heterogéneo engloba muchos ítems con diferentes características y, con mayor importancia, diferentes metas que deben satisfacer. Un espacio de recomendación homogéneo implica que el conocimiento del contenido relativo al dominio se adquirirá y mantendrá de forma más simple.
- **Riesgo:** Los dominios de recomendación se pueden distinguir por el grado de riesgo que toma un usuario al aceptar una recomendación, siendo, por ejemplo, 0.99\$ por una canción un costo bajo y un diagnóstico médico un riesgo alto. El riesgo determinará la tolerancia del usuario hacia los falsos positivos y negativos que otorgue el SR.
- **Rotación:** Los SR se utilizan en dominios con ítems de larga vida útil, como libros, así como en dominios donde el valor o la relevancia de un ítem se degrada rápidamente,

como noticias. Un dominio con alta rotación es aquél en el que los ítems vienen y van de forma rápida, y el SR encara un flujo continuo de ítems nuevos para integrar a sus fuentes de datos. Los dominios con alta rotación incrementan la dispersión de los datos de opinión ya que los ítems nuevos no han sido observados por muchos usuarios y los ítems que han acumulado muchas calificaciones dejan de ser relevantes.

- **Estilo de interacción:** Las interacciones con el SR pueden ser implícitas o explícitas. Las entradas implícitas tienen un error natural ya que son inferidas del comportamiento del usuario, y su mejor uso es recopilar información de calificaciones simples. Las entradas explícitas pueden ser mas dispersas si el costo de generarlas para el usuario es relativamente alto; se deberían plantear estrategias para que los usuarios perciban beneficios que justifiquen el esfuerzo.
- **Estabilidad de las preferencias:** Las preferencias del usuario también tienen grados de duración variables, pudiendo el interés de un comprador de un ítem decrecer posterior a la compra, o subir y bajar de forma cíclica de acuerdo a temporadas deportivas. Tener preferencias estables implica que es probable que los datos recolectados en el pasado aún sean válidos hoy. Tener preferencias inestables significa que es posible que cualquier dato recolectado en el pasado tenga que ser descartado, lo que incrementa la importancia de reunir los requerimientos específicos del usuario en el momento en que usa el SR. El problema de preferencias inestables puede ser aliviado almacenando más datos; si un usuario genera suficiente data de opinión durante una sesión para representar adecuadamente sus preferencias actuales no existirá la necesidad de extrapolarla de datos históricos.
- **Capacidad de explicación:** Ciertas aplicaciones, por ejemplo aquellas de alto riesgo, pueden requerir que el sistema pueda explicar sus recomendaciones. Dichas explicaciones mejoran la confianza de que la recomendación es apropiada e incrementan la probabilidad de que la misma sea aceptada. La explicación de recomendaciones es más sencilla cuando se utilizan fuentes de conocimiento del contenido, mientras que el problema se hace más complejo para fuentes de conocimiento sociales.

II.1.7.3 Fuentes de conocimiento

Así como el dominio y las características de la aplicación colocan ciertas restricciones en los tipos de fuentes de conocimiento que un SR puede utilizar, la disponibilidad y calidad de cada fuente de conocimiento influencia que tecnologías de recomendación puede utilizar un SR de manera provechosa (Burke & Ramezani, 2010).

II.1.7.3.1 Conocimiento social

El conocimiento social permite el uso de algoritmos colaborativos en los que se extrapolan predicciones individuales de las opiniones de sus pares, siendo las calificaciones el tipo de datos más utilizado para modelar este conocimiento. Las calificaciones son frecuentemente conceptualizadas como matrices $m \times n$ donde m es el número de usuarios y n el

número de ítems y cada entrada corresponde a la calificación del usuario para el ítem. Las técnicas basadas en modelos utilizan esta matriz para crear un modelo con antelación, mientras que las técnicas basadas en memoria la utilizan en el momento de generar la recomendación para producir la predicción.

En dominios heterogéneos, el conocimiento social debería ser considerado una fuente de conocimiento ya que es recopilado de las entradas del usuario y no necesita adaptaciones extensivas. Sin embargo, el conocimiento social no es suficientemente preciso y confiable para dominios con alto riesgo o que requieren explicación de las recomendaciones. Adicionalmente, el conocimiento social tiende a ser disperso para dominios con alta rotación, ya que disminuye la probabilidad de que cualquier usuario tenga la oportunidad de calificar un ítem dado, aunque esto puede ser mitigado teniendo una población de usuarios suficientemente amplia. Finalmente, es apropiado utilizar el conocimiento social en dominios de interacción implícita, ya que es posible minar el comportamiento de los usuarios utilizando técnicas estadísticas y de aprendizaje de máquina, que son típicas en los algoritmos de filtrado colaborativo. En dominios con preferencias de usuario inestables el conocimiento social puede ser engañoso ya que los datos históricos no son tan confiables (Burke & Ramezani, 2010).

II.1.7.3.2 Conocimiento individual

El conocimiento individual es un requerimiento esencial para que un SR produzca recomendaciones personalizadas. En recomendaciones colaborativas el conocimiento individual del usuario se corresponde con el conocimiento social tomado de la población general de usuarios, realizándose de su forma más simple cuando las fuentes son del mismo tipo, requiriéndose únicamente una métrica de similitud para comparar individuos.

Puede ser difícil transferir las entradas del usuario en ciertos ítems para recomendar otros cuando el dominio es heterogéneo. A su vez, es más probable que alguna forma de recomendación basada en conocimiento sea adecuada para un dominio que requiera conocimiento de los requerimientos a corto plazo del usuario. En dominios de alto riesgo y dominios que requieren explicación es usualmente necesario tener requerimientos y restricciones explícitas del usuario. Similarmente, es más probable que se necesiten requerimientos del usuario en dominios con preferencias inestables (Burke & Ramezani, 2010).

II.1.7.3.3 Conocimiento del contenido

La forma más básica de conocimiento del contenido son las características de los ítems, el tipo de datos que generalmente están disponibles en una base de datos de productos. Estas características pueden, generalmente, ser usadas en su forma original en un SR, aunque el diseñador del mismo debería tratar de restringir el espacio de las características a aquellas más relevantes.

Si los ítems son representados por documentos no estructurados, como noticias, el implementador necesitará tomar herramientas de extracción de información para seleccionar y extraer las características a ser usadas en la recomendación. Algunas técnicas estándar incluyen eliminar las palabras vacías¹ o técnicas de selección más sofisticadas como divergencia de Kullback-Leibler, información mutua o entropía cruzada. Los documentos más estructurados, como paginas HTML, ofrecen más oportunidades para extraer características (Burke & Ramezani, 2010).

La calidad de las recomendaciones producidas por un SR basado en contenido o basado en conocimiento son completamente dependientes de la calidad de los datos del contenido en que se basan las decisiones, a tal punto, que la falta de características de ítems confiables es, frecuentemente, el factor decisivo para evitar recomendaciones basadas en contenido. El costo de crear y mantener una base de datos de características de ítems útiles es alto, especialmente en dominios heterogéneos. Si existe un alto número de características no independientes extraídas de varias formas el sistema puede tolerar datos de características con errores, mientras que las aplicaciones de alto riesgo deben prestar atención especial a tener características de ítem limpias y, típicamente, se requerirá de revisiones manuales de los datos de las mismas (Burke & Ramezani, 2010).

II.1.7.3.3.1 Conocimiento del dominio

Un SR basado en conocimiento necesita, típicamente, tener conocimiento adicional a las características asociadas a los ítems. La forma más básica de conocimiento del dominio que un SR puede utilizar es una ontología de las características de los ítems, tal que permita al sistema operar sobre las relaciones entre características a un nivel más profundo que solo igualdad o diferencia.

Muchas elecciones de alto riesgo traen consigo restricciones impuestas por el dominio que un SR debe obedecer. El problema de recomendación puede, en algunos casos, formularse completamente como una satisfacción de restricciones con restricciones contribuidas tanto por el usuario como por el sistema.

Una categoría final de conocimiento del dominio es el conocimiento de medios y fines, el que le permite al sistema realizar un plan entre las metas del usuario (los fines) y los ítems que pueden satisfacerlo (los medios). Parte de la razón por la cual los usuarios se benefician de los SR es que pueden tomar buenas decisiones sin necesariamente familiarizarse con todas las complejidades del espacio de los ítems. La tabla 1 resume estas consideraciones de dominio y su impacto en las fuentes de conocimiento.

¹ Palabras sin significado como artículos, pronombres, proposiciones, etc.

Característica	Social	Individual	Contenido
Heterogéneo		Puede tener transferencia pobre a ítems no vistos	Difícil de diseñar y mantener
Alto riesgo	No suficientemente preciso / confiable	Usualmente se necesitan requerimientos y restricciones	Se necesitan restricciones del dominio
Alta rotación	Datos dispersos		
Implícito		Requerimientos detallados no disponibles	
Preferencias inestables	Datos históricos no confiables	Datos históricos no confiables Se necesitan los requerimientos del usuario	
Necesidad de explicaciones	Explicaciones débiles	Se pueden relacionar los requerimientos con los ítems	Se puede usar conocimiento del dominio

Tabla 1: Impacto de dominios de recomendación en fuentes de conocimiento (Burke & Ramezani, 2010)

II.1.7.4 Relacionando dominios con tecnologías

Se deben tener en cuenta algunas consideraciones básicas al considerar el dominio de recomendación. Primero, existen algunos tipos de dominios para los cuales el conocimiento social no parece ser muy útil, en particular dominios de alto riesgo y de alta rotación, en los que no parece que haya suficiente tiempo para que un ítem construya una reputación entre un alto número de usuarios pares antes de ser reemplazado con otros ítems.

Cuando un dominio tiene un alto riesgo asociado, la mayoría de los usuarios necesitan una explicación convincente de lo apropiado de una recomendación mas allá de que otros usuarios la hayan utilizado. Incluso un perfil aprendido de las interacciones previas del usuario puede no ser aceptable si al adherirse al mismo se obvian consideraciones cruciales al contexto actual.

Similarmente, observando la interacción, se ve que no siempre es posible recolectar cada tipo de conocimiento de cada tipo de interacción. En los sistemas con entradas implícitas no se recolecta ningún tipo de requerimiento directo del usuario, aunque algunas veces sea posible extraer requerimientos explícitos de su actividad.

La inestabilidad de preferencias favorece las técnicas basadas en conocimiento. Aprender de las interacciones previas del usuario puede resultar en un obstáculo en vez de una ayuda. Sin embargo, en algunos casos, como en la personalización web, los usuarios pueden proveer suficientes datos implícitos en una sesión para formar un perfil útil que puede ser comparado a otros.

En casos en los que los criterios no ayudan a alcanzar una conclusión definitiva, es válido notar que las diferentes tecnologías tienen diferentes costos de implementación y mantenimiento. Es probable que las recomendaciones colaborativas sean las menos costosas de implementar, ya que requieren una base de datos de calificaciones pero no requieren características limpias y bien diseñadas, que son el requerimiento mínimo para las otras tecnologías de recomendación. Las tecnologías basadas en conocimiento serán las más costosas ya que requieren ingeniería del conocimiento y mantenimiento continuo (Burke & Ramezani, 2010). La tabla 2 muestra la influencia de los diferentes factores del dominio en la elección del enfoque de recomendación.

Factor		Colaborativo	Basado en Contenido	Basado en Conocimiento
Heterogéneo	Bajo	✓	✓	✓
	Alto	+	-	-
Riesgo	Bajo	✓	✓	✓
	Alto	-	-	+
Rotación	Baja	✓	✓	✓
	Alta	-	+	+
Interacción	Implícita	+	+	-
	Explícita	✓	✓	+
Preferencias	Estables	✓	✓	✓
	Inestables	-	-	+
Explicaciones	Requeridas	-	-	+
	No requeridas	✓	✓	✓

Tabla 2: Factores del dominio y técnicas de recomendación (Burke & Ramezani, 2010)

II.1.7.4.1 Algoritmos de filtrado colaborativo

En las recomendaciones colaborativas basadas en usuario, un subconjunto de usuarios apropiados son escogidos de acuerdo con su similitud al usuario activo y se utiliza un agregado normalizado de las calificaciones para generar predicciones para el usuario activo en tiempo de ejecución. Diferentes implementaciones de filtrado colaborativo aplican variaciones de algoritmos de predicción basados en vecindad y, en general, se toman como válidos los siguientes lineamientos (Herlocker, Konstan, Borchers, & Riedl, 1999):

- Si la escala de calificaciones consiste de un pequeño número de rangos discretos (por ejemplo, enteros del 1 al 5 o del 1 al 20) utilizar el coeficiente de correlación de Spearman como la medida de similitud.
- Si la escala de calificaciones no es discreta sino continua, se recomienda considerar el coeficiente de correlación de Pearson.
- Si la escala de calificación es binaria o unaria, se recomienda considerar un enfoque diferente, como el uso de redes bayesianas.
- Es importante utilizar una ponderación por significancia para devaluar los correlados con números bajos de ítems co-calificados ya que, frecuentemente, aportaran un mayor aumento en precisión que otros algoritmos de similitud.
- Los usuarios calificarán los ítems en escalas ligeramente diferentes, por lo que se recomienda utilizar la desviación de la media para ponderar las mismas.

El filtrado colaborativo basado en ítems es un algoritmo basado en memoria que explora la relación entre ítems como una función de cómo los han calificado los usuarios. Se ha mostrado que la versión basada en ítems del algoritmo *k-vecinos* escala mejor y produce recomendaciones más precisas que las basadas en usuarios para conjuntos de datos grandes. Sin embargo, los algoritmos basados en memoria de vecino más cercano tienen dos límites computacionales: escala y dispersión. La necesidad de comparar cada usuario contra cada otro usuario hace estas técnicas poco eficientes para conjuntos grandes. Adicionalmente, la necesidad de comparar calificaciones de ítems implica que, en conjuntos muy dispersos, los usuarios tendrán muy pocos vecinos (Burke & Ramezani, 2010).

Las técnicas de reducción de dimensionalidad (por vías de descomposición de valor singular, análisis semántico latente u otras técnicas) son para este momento un enfoque estándar para sobrellevar la dispersión en las bases de datos de calificaciones. Las varias formas de compresión y/o reducción de dimensionalidad usualmente requieren cálculos extensivos *offline*, pero como resultado escalan mucho mejor (Burke & Ramezani, 2010).

II.2 Algoritmo de predicción *Slope One*

El algoritmo de predicción para filtrado colaborativo en línea *Slope One* (pendiente uno) se basa en el principio intuitivo de un diferencial de popularidad entre ítems para usuarios. El algoritmo determina, de par en par, cuanto más agradable es un ítem que otro y utiliza esa diferencia para predecir calificaciones.

Slope One trata de proveer planes robustos que cumplan con las siguientes cinco metas (Lemire & Maclachlan, 2005):

- Facilidad de implementar y mantener: todos los datos agregados deberían ser interpretados fácilmente por el desarrollador promedio y los algoritmos deberían ser fáciles de implementar y probar.

- Actualizaciones en caliente: la adición de una nueva calificación debería cambiar todas las predicciones de forma instantánea.
- Eficiencia en momento de consulta: las consultas deberían ser rápidas, posiblemente a costa del almacenamiento.
- Pocas expectativas de usuarios nuevos: los usuarios con pocas calificaciones deberían recibir recomendaciones válidas.
- Precisión razonable: los planes deberían ser competitivos con los sistemas más precisos, pero no siempre vale la pena sacrificar en simplicidad o escalabilidad para generar una ganancia menor en precisión.

Una forma en la que se puede medir el diferencial de popularidad entre dos ítems dados es simplemente substrayendo la calificación promedio de los dos ítems y, posteriormente, utilizando la diferencia para predecir la calificación de otro usuario de uno de esos ítems, dada la calificación del otro. Si se consideran dos usuarios *A* y *B* y dos ítems *I* y *J* tal que el usuario *A* ha asignado al ítem *I* una calificación de 1, mientras que el usuario *B* le asignó una calificación de 2, se observa que el ítem *J* está calificado más que el ítem *I* por 0,5 puntos, por lo que se podría predecir que el usuario *B* dará al ítem *J* una calificación de 2,5. La figura 12 ilustra este proceso:

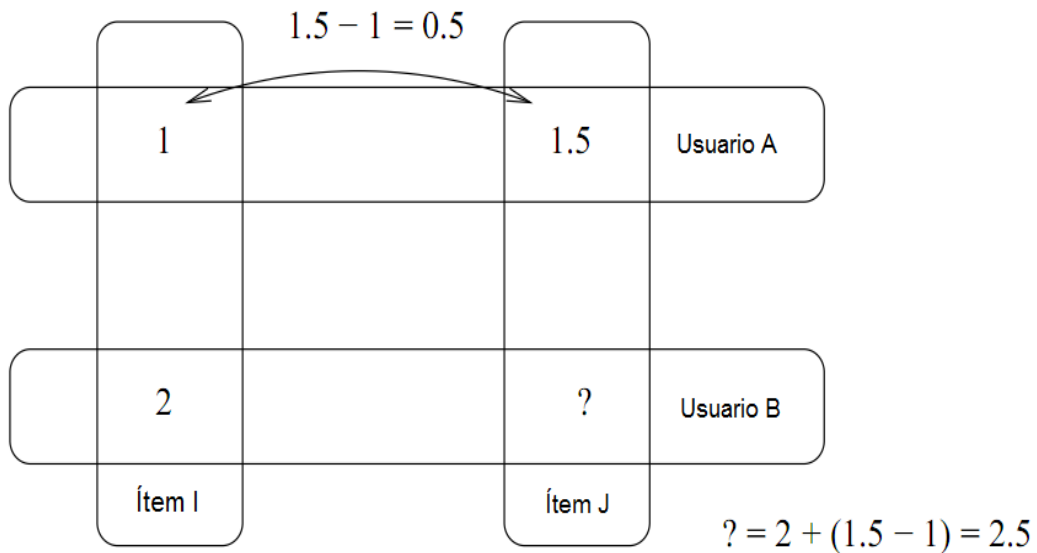


Figura 12: Base de los planes Slope One: las calificaciones del usuario A de dos ítems y del usuario B de un ítem común se utiliza para predecir la calificación desconocida del usuario B (Lemire & Maclachlan, 2005).

Los predictores de Slope One toman la forma $f(x) = x + b$ donde b es una constante y x es una variable que representa valores de calificaciones. Para cualquier par de ítems, se trata de encontrar la mejor función f que prediga las calificaciones de un ítem basado en las calificaciones de los otros ítems, siendo f una función diferente para cada par de ítems. En la

publicación de Slope One se proponen tres planes de filtrado colaborativo: Slope One básico, Slope One ponderado y Slope One bipolar.

II.2.1 Slope One básico

Los planes basados en Slope One toman en cuenta la información de otros usuarios que han calificado el mismo ítem y la información de los otros ítems calificados por el mismo usuario. Adicionalmente, se toma en cuenta puntos de datos que no caen en el arreglo de los usuarios ni en el arreglo de los ítems, como la calificación del ítem i dada por el usuario A en la figura 11. Gran parte de la fuerza del enfoque viene de que únicamente las calificaciones hechas por usuarios que han calificado algún ítem común con el usuario que realiza la predicción y las calificaciones de ítems que el mismo también ha calificado entran en la predicción de calificaciones bajo SR basados en Slope One. Esta implementación no depende de cómo califica el usuario los ítems individuales, sino de las calificaciones promedio del usuario y, crucialmente, a cuales ítems ha calificado el mismo (Lemire & Maclachlan, 2005).

Formalmente, dados dos arreglos de evaluación v_i y w_i , con $i = 1, \dots, n$, se busca el mejor predictor de la forma $f(x) = x + b$ para predecir w dado v minimizando $\sum_i (v_i + b - w_i)^2$. Al derivar con respecto a b e igualar la derivada a cero, se consigue que $b = \frac{\sum_i w_i - v_i}{n}$. En otras palabras, se debe escoger la constante b igual a la diferencia promedio entre los dos arreglos. Dado un conjunto de entrenamiento X y dos ítems cualquiera i y j con calificaciones u_j y u_i respectivamente en alguna evaluación de usuario u (connotada como $u \in S_{j,i}(X)$), se considera la desviación promedio del ítem i con respecto al ítem j por la fórmula representada en la figura 13, siendo $\text{card}(S_{j,i}(X))$ la cardinalidad del conjunto $S_{j,i}(X)$.

$$\text{dev}_{j,i} = \sum_{u \in S_{j,i}(X)} \frac{u_j - u_i}{\text{card}(S_{j,i}(X))}$$

Figura 13: Fórmula para calcular la desviación media del ítem i con respecto al ítem j (Lemire & Maclachlan, 2005).

La matriz simétrica definida por $\text{dev}_{j,i}$ puede ser calculada una vez y actualizada rápidamente cuando se ingresan nuevos datos. Dado que $\text{dev}_{j,i} + u_i$ es una predicción para u_j dado u_i , un predictor razonable puede ser el promedio de todas esas predicciones. El algoritmo Slope One utiliza una aproximación para calcular esta medida, dada por la fórmula de la figura 14, siendo $R_j = \{i \mid i \in S(u), i \neq j, \text{card}(S_{j,i}(X)) > 0\}$:

$$p^{S1}(u)_j = \sum_{i \in R_j} \left(\frac{u_j}{\text{card}(R_j)} \right) + \frac{1}{\text{card}(R_j)} \sum_{i \in R_j} (\text{dev}_{j,i} + u_i)$$

Figura 14: Fórmula para calcular una predicción de calificación del usuario u para el ítem j usando Slope One (Lemire & Maclachlan, 2005).

II.2.2 Slope One ponderado

Una de las desventajas de Slope One es que el número de calificaciones observadas no se toma en consideración. Intuitivamente, para predecir la calificación del usuario A para el ítem L dada la calificación del usuario A para los ítems J y K , si 2000 usuarios calificaron el par de ítems J y L y sólo 20 usuarios calificaron el par de ítems K y L , entonces es probable que la calificación del usuario A para el ítem J sea un mejor predictor para el ítem L que la calificación del usuario A para el ítem K . Por lo tanto, se define la predicción Slope One ponderada por la formula representada en la figura 15, donde $c_{i,j} = \text{card}(S_{j,i}(X))$.

$$p^{wS1}(u)_j = \frac{\sum_{i \in S(u) - \{j\}} (\text{dev}_{j,i} + u_i) c_{j,i}}{\sum_{i \in S(u) - \{j\}} c_{j,i}}$$

Figura 15: Fórmula para calcular una predicción de clasificación del usuario u para el ítem j usando Slope One ponderado (Lemire & Maclachlan, 2005).

II.2.3 Slope One bipolar

Si bien ponderar funciona para favorecer patrones de calificaciones que ocurren frecuentemente sobre aquellos ítems más infrecuentes, Slope One Bipolar separa la predicción en dos partes, utilizando Slope One Ponderado para calcular una predicción en base a los ítems que gustaron a los usuarios y otra predicción utilizando los ítems que disgustaron a los usuarios.

Dada una escala de calificaciones, por ejemplo de 0 a 10, podría parecer razonable utilizar el medio de la escala, 5, como el umbral para definir cuales ítems gustan y cuales disgustan. Esto funcionaría si las calificaciones de los usuarios están distribuidas uniformemente, sin embargo, éste no es el caso en los conjuntos de datos de prueba típicos. Para soportar usuarios balanceados, optimistas, pesimistas y bimodales se aplica el promedio de las calificaciones del usuario como margen entre los ítems agradados y no agradados para ese usuario (Lemire & Maclachlan, 2005).

Con respecto a la figura 11, para generar la predicción para el ítem J del usuario B , Slope One Bipolar restringe el conjunto de calificaciones que son predictivas más que en las versiones anteriores: con respecto a los ítems, sólo desviaciones entre dos ítems gustados o disgustados se toman en consideración y, con respecto a los usuarios, solo las desviaciones de pares de

usuarios que calificaron tanto I como J y que comparten un gusto o disgusto por I se utilizan para predecir las calificaciones para J .

Esta fisión de cada usuario en gustos y disgustos efectivamente dobla el número de usuarios. Sin embargo, las restricciones establecidas reducen el número total de calificaciones en el cálculo de las predicciones. Formalmente, se divide cada evaluación de u en dos conjuntos de ítems calificados: $S^{gusto}(u) = \{i \in S(u) | u_i > \bar{u}\}$ y $S^{disgusto}(u) = \{i \in S(u) | u_i < \bar{u}\}$. Por cada par de ítems i, j se divide el conjunto de todas las evaluaciones X en $S_{i,j}^{gusto} = \{u \in X | i, j \in S^{gusto}(u)\}$ y $S_{i,j}^{disgusto} = \{u \in X | i, j \in S^{disgusto}(u)\}$. Utilizando estos dos conjuntos, se calculan las matrices de desviaciones para los ítems agradados, con la fórmula representada en la figura 16, y para los ítems no agradados, con su fórmula equivalente utilizando los conjuntos respectivos (Lemire & Maclachlan, 2005).

$$dev_{j,i}^{gusto} = \sum_{u \in S_{j,i}^{gusto}(X)} \frac{u_j - u_i}{card(S_{j,i}^{gusto}(X))}$$

Figura 16: Cálculo de la desviación media del ítem i con respecto al ítem j para ítems agradados (Lemire & Maclachlan, 2005).

La predicción para la calificación del ítem j basado en la calificación del ítem i es, entonces, $dev_{j,i}^{gusto}$ más la calificación que el usuario da para el ítem i o $dev_{j,i}^{disgusto}$ más dicha calificación u_i , dependiendo si i pertenece a $S^{gusto}(u)$ o $S^{disgusto}(u)$. La predicción Slope One ponderada es definida por la fórmula de la figura 17, donde los ponderados $c_{j,i}^{gusto} = card(S_{j,i}^{gusto})$ y $c_{j,i}^{disgusto} = card(S_{j,i}^{disgusto})$.

$$P^{bps1}(u)_j = \frac{\sum_{i \in S^{gusto}(u) - \{j\}} p_{j,i}^{gusto} c_{j,i}^{gusto} + \sum_{i \in S^{disgusto}(u) - \{j\}} p_{j,i}^{disgusto} c_{j,i}^{disgusto}}{\sum_{i \in S^{gusto}(u) - \{j\}} c_{j,i}^{gusto} + \sum_{i \in S^{disgusto}(u) - \{j\}} c_{j,i}^{disgusto}}$$

Figura 17: Fórmula para calcular una predicción de clasificación del usuario u para el ítem j usando Slope One Bipolar ponderado (Lemire & Maclachlan, 2005).

II.2.4 Resultados Experimentales

Los autores de Slope One utilizan el MAE de todos menos uno, en el cual, se esconden calificaciones de forma sucesiva e individual en el conjunto de prueba y se realiza una predicción de la calificación escondida, computando el error promedio de la predicción. Los algoritmos Slope One fueron probados con dos conjuntos de prueba utilizados frecuentemente para

evaluar enfoques de filtrado colaborativo: EachMovie, hecho disponible por Compaq Research, y el conjunto de datos Movielens del grupo Movielens de la universidad de Minnesota. Los datos son reunidos de sitios web de calificaciones de películas donde las calificaciones van de 0,0 a 1,0 en incrementos de 0,2 en el caso de EachMovie, y de 1 a 5 en incrementos de 1 en el caso Movielens. Se tomaron 50.000 calificaciones como un conjunto de entrenamiento X y un conjunto adicional de evaluaciones con un total de al menos 100.000 calificaciones para el conjunto de prueba X' . La tabla 3 resume los resultados para las varias opciones evaluadas utilizando el mismo conjunto de datos y la misma métrica:

Plan	EachMovie	Movielens
Slope One Bipolar	0,194	0,188
Slope One Ponderado	0,198	0,188
Slope One	0,200	0,188
Coseno ajustado basado en ítems	0,209	0,198
Promedio por usuario	0,231	0,208
Correlación de Pearson	0,194	0,190

Tabla 3: MAE de todos menos uno para los conjuntos de datos EachMovie y Movielens de cada esquema, menor es mejor (Lemire & Maclachlan, 2005).

Los resultados mostraron que el algoritmo Slope One básico tuvo mejores resultados que los esquemas clásicos de coseno ajustado y promedio por usuario. La modalidad Slope One Bipolar mejora los resultados alrededor de 2% en el conjunto EachMovie y logra una precisión comparable a la del, más complejo, esquema basado en la correlación de Pearson.

II.2.5 Algoritmo de predicción

Las tres variantes de Slope One utilizan, a gran escala, el mismo algoritmo general, que se puede expresar de la siguiente forma en lenguaje natural:

1. Fase de pre-procesamiento: se calculan las diferencias entre las calificaciones de los ítems
2. Para cada par de ítems i y j buscar los usuarios u que califican ambos ítems.
3. Para cada usuario u , añadir la diferencia en las preferencias para i y j a un promedio.
4. Almacenar los promedios calculados en el paso anterior.
5. Fase de recomendación:

6. Para cada ítem i que el usuario u no ha calificado y cada ítem j que el usuario ha calificado, buscar la diferencia promedio entre i y j
7. Anadir la diferencia hallada en el paso anterior a la calificación de u para j
8. Anadir esta diferencia a un promedio de ejecución.
9. Generar la recomendación: retornar un ítem, o un número pequeño de ítems, ordenados por los promedios calculados durante la fase 2.

La fase 1 se puede realizar *offline*, almacenando los resultados de la misma como una matriz de $M \times M$, frecuentemente denominada matriz ítems-a-ítems, donde M es la cantidad de ítems del sistema. Cada vez que se ingresa una nueva calificación se actualiza esta matriz, aunque dependiendo de la implementación este proceso se realiza en línea o *offline*. Luego, para generar la recomendación, basta definir una función de predicción, de acuerdo a la versión de Slope One seleccionada, que prediga la calificación del usuario actual para cada ítem y ejecutarla de forma iterativa (Lemire & McGrath, 2005).

II.2.6 Implementaciones

El algoritmo Slope One, en sus tres variantes, se ha utilizado en varios sistemas de recomendación, entre los que destacan:

- Hitflip (<http://www.hitflip.de/>), un sitio de intercambio de películas, libros, juegos y música en alemán.
- RACOFI (<http://www.nrc-cnrc.gc.ca/eng/ibp/iit.html>), un sistema genérico de recomendación realizado por el National Research Council de Canadá.
- Value Investing News (<http://www.valueinvestingnews.com>), un sitio de noticias del mercado de valores, en el que se recomiendan noticias al usuario.
- Qué libro me recomiendas (<http://quelibromerecomiendas.com/>), un sistema de recomendación de libros en español.
- AllFamo (<http://www.allfamo.com>), un motor de recomendación comercial para comercio electrónico y medios sociales.
- Sopa de Libros (<http://www.sopadelibros.com/>), una red social de lectura en español.

Todos estos SR se basan en calificaciones explícitas de los usuarios para generar las tablas de desviación. Algunos, como Value Investing News o Qué libro me recomiendas, ofrecen la posibilidad de recomendaciones generales, aunque todos se enfocan en las recomendaciones personalizadas para un usuario. Similarmente, todos los sistemas presentan las recomendaciones al usuario en la forma de listas de las N -tope con mayor calificación predicha.

Adicionalmente, el algoritmo Slope One ha sido implementado en una gran cantidad de lenguajes de programación, resultando en una gran cantidad de librerías de código abierto, entre las que destacan:

- Implementación en Python por Bryan O'Sullivan (<http://www.serpentine.com/blog/2006/12/12/collaborative-filtering-made-easy/>),.

- Implementación en Java por Daniel Lemire (<http://lemire.me/fr/documents/publications/SlopeOne.java>).
- Implementación en PHP/SQL por Daniel Lemire y Sean McGrath (Lemire & McGrath, 2005).
- MyMediaLite (<http://www.ismll.uni-hildesheim.de/mymedialite>), una librería multipropósito de algoritmos de recomendación escrita en C# para la plataforma .NET.
- OpenSlopeOne (<http://code.google.com/p/openslopeone/>), una Librería de código abierto basada en PHP y MySQL.
- LensKit (<http://lenskit.grouplens.org>), un *toolkit* de código abierto para construir, evaluar e investigar sistemas de recomendación.

Algunas de estas implementaciones son primordialmente un ejercicio de aprendizaje y otorgan instrucciones y recomendaciones para implementar Slope One en cada lenguaje de programación, como la implementación de Bryan O'Sullivan, que ha sido utilizada como base para muchas otras implementaciones por el detalle de la explicación anexa. Otras, como MyMediaLite y OpenSlopeOne, son librerías que permiten utilizar Slope One en un sistema de forma rápida sin preocuparse de la implementación interna del algoritmo.

II.3 Integración con sitios de redes sociales

Desde su introducción, los sitios de redes sociales (SRS) como Facebook, Twitter o Google+, entre otros, han atraído millones de usuarios, muchos de los cuales han integrado estos sitios en sus prácticas diarias. Actualmente existe un gran número de SRS soportando un gran rango de intereses y prácticas. Si bien la cultura emergente de cada SRS es variada, las características tecnológicas clave son relativamente constantes.

II.3.1 Definición

Boyd y Ellison (2007) definen los SRS como servicios basados en la web que permiten a los individuos construir un perfil público, o semi-público, dentro de un sistema, articular una lista de usuarios con los que el usuario comparte una conexión, y ver y recorrer su lista de conexiones y las de otros usuarios dentro del sistema, variando la naturaleza y la nomenclatura de estas conexiones entre varios sitios.

Los SRS no sólo permiten a los individuos articular y hacer visibles sus redes sociales, lo que puede resultar en conexiones entre individuos que no se realizarían normalmente. Sin embargo, ésta no es la meta, y las conexiones se realizan frecuentemente entre individuos con alguna relación previa. En muchos de los SRS mas grandes, los participantes no están necesariamente buscando conocer nuevos individuos, sino comunicándose principalmente con personas que ya son parte de su red social extendida.

II.3.2 Características

Si bien los SRS han implementado una gran variedad de características técnicas, su eje central consiste de perfiles visibles que muestran una articulada lista de amigos que también son usuarios del sistema. Teniendo en cuenta la definición previa, existen tres características básicas en la construcción de un SRS: perfiles, lista de amigos y herramientas para comunicación pública.

II.3.2.1 Perfiles

Después de unirse a un SRS, se le solicita al usuario llenar formas que contienen preguntas posteriormente utilizadas para generar el perfil; dichas preguntas típicamente incluyen descriptores como edad, ubicación, intereses y una sección general, y la mayoría de los SRS incentivan al usuario a proveer una foto. Algunos sitios permiten a los usuarios agregar contenido multimedia a su perfil o modificar el aspecto del mismo, mientras que otros permiten a los usuarios agregar módulos o aplicaciones para mejorar su perfil.

La visibilidad de un perfil varía de sitio a sitio y de acuerdo a la discreción del usuario. Algunos SRS permiten que sus perfiles sean leídos por los motores de búsqueda, haciéndolos públicos, otros controlan lo que puede ver un usuario de acuerdo al estatus de su cuenta en el SRS. Otra opción es permitir a los usuarios decidir si su perfil es público o sólo puede ser visto por sus amigos, o tomar por definición que los usuarios que son parte de la misma red social pueden ver los perfiles del otro hasta ser indicado lo contrario. Las variaciones estructurales alrededor de la visibilidad y acceso al perfil son una de las principales formas en las que los SRS se diferencian de sus competidores (Boyd & Ellison, 2007).

En adición a ser un sitio de auto representación, los perfiles son un lugar donde los usuarios se reúnen a conversar y compartir. Las conversaciones suceden en los perfiles y el perfil de un usuario refleja su dedicación al SRS. Como resultado, los participantes no tienen un completo control sobre su auto representación. Aunque ciertas características pueden permitir a los participantes restringir las contribuciones a su perfil, la mayoría de los mismos reciben positivamente la contribución de imágenes, videos y comentarios (Boyd, 2011).

II.3.2.2 Listas de amigos

Luego de unirse a un SRS se solicita a los usuarios que identifiquen a otros usuarios en el sistema con quienes se tiene una relación. La denotación para estas relaciones difiere dependiendo del sitio, con términos populares como amigos, contactos o fans. La mayoría de los SRS requieren confirmaciones bidireccionales de las relaciones, aunque algunos utilizan también lazos unidireccionales, frecuentemente etiquetados como fans o seguidores (Boyd & Ellison, 2007).

La visualización pública de conexiones es un componente crucial de los SRS. La lista de amistades contiene enlaces al perfil de cada amigo, permitiendo al usuario recorrer el grafo de

la red haciendo clics en las listas de amigos. En la mayoría de los sitios la lista es visible para cualquiera que tenga permisos para ver el perfil, aunque existen excepciones en las que se permite al usuario ocultar su red (Boyd, 2011).

El listado de amigos es tanto social como político, ya que al escoger quien se incluye como amigo, los participantes generalmente consideran las implicaciones de excluir o explícitamente rechazar a alguien en lugar de los beneficios de incluirlos. Si bien existen participantes que controlaran estrictamente su lista de amigos y participantes que gregariamente agregarán a cualquier usuario, la mayoría de los mismos simplemente incluye a todo aquél que considere una parte de su mundo social. Esto puede incluir a amigos y conocidos actuales y pasados así como relaciones secundarias, o personas que el usuario apenas conoce pero se siente obligado a incluir (Boyd, 2011).

Una forma de interpretar esta lista pública de conexiones es observarla como una audiencia, ya que estos amigos son las personas con los que los participantes se ven conectando masivamente. Para algunos participantes es importante asegurarse que estos todos estos individuos son parte del mismo contexto social, mientras que, para otros, mezclar contextos sociales diferentes es aceptable y deseable. Cómo un participante ataca los diferentes contextos sociales da forma a quienes incluirá o no como amigos (Boyd, 2011).

II.3.2.3 Herramientas de comunicación pública

La mayoría de los SRS también proveen un mecanismo para que los usuarios dejen mensajes en los perfiles de sus amigos. Esta característica típicamente involucra dejar comentarios, aunque los sitios emplean varias etiquetas para esta característica. Adicionalmente, los SRS frecuentemente poseen una característica de mensajería privada similar al correo web. Si bien los mensajes, tanto privados como públicos, son populares en la mayoría de los SRS, no están disponibles universalmente.

En conjunción con los comentarios, muchos SRS implementan funcionalidades que permiten a los participantes difundir contenido a todos sus amigos. Estas funcionalidades permiten a los usuarios contribuir contenido que luego es distribuido a los amigos principalmente mediante actualizaciones automáticas provenientes de cada amigo. En algunos casos, estas actualizaciones son mostradas nuevamente en el perfil del usuario y permiten realizar comentarios al respecto. Mientras que las actualizaciones individuales son posiblemente mundanas, la corriente de contenido le da a los participantes una sensación general de aquellos que los rodean (Boyd, 2011).

Más allá de perfiles, amigos, comentarios y mensajes privados, los SRS varían de gran forma en sus características y base de usuarios. Algunos tienen capacidades para compartir fotos o video, otros tienen blogs incluidos y soporte para mensajería instantánea. Existen SRS específicos para tecnologías móviles y existen SRS basados en web que soportan interacciones móviles completa o limitadamente. Muchos SRS se enfocan en grupos de personas de regiones

o grupos lingüísticos específicos, e incluso algunos sitios están diseñados con orientaciones políticas, religiosas, étnicas o sexuales específicas (Boyd & Ellison, 2007).

II.4 Conclusiones

En base a la investigación realizada, se observa que el enfoque de recomendación de filtrado colaborativo es adecuado para el problema planteado. No se recomienda el uso de un SR basado en conocimiento por la falta de conocimiento del dominio y la dificultad de procesar el lenguaje natural para generar modelos que permitan su utilización. Similarmente, no se recomienda el uso de un SR basado en contenido por la dificultad para categorizar automáticamente el contenido.

La familia de algoritmos *Slope One* es la recomendada como producto de la investigación, debido entre otras características a que permite la generación y actualización de preferencias en caliente sin sacrificar precisión o simplicidad, es idónea para el ambiente *Web* y posee un bajo costo de implementación y mantenimiento asociado.

CAPÍTULO III: MARCO METODOLÓGICO

El desarrollo de la aplicación planteada para resolver el problema de investigación involucrará la utilización de una metodología *ad-hoc* basada en el desarrollo iterativo e incremental.

III.1 Desarrollo Incremental e Iterativo

El desarrollo incremental se basa en desarrollar y desplegar las varias partes del sistema de forma separada, teniendo como intención que el proyecto pueda adaptarse a la retroalimentación externa. El desarrollo iterativo, por otra parte, involucra incluir tiempo para revisar y mejorar las partes del sistema. En un desarrollo incremental e iterativo el proyecto se planifica en diversos bloques temporales, en los cuales se evoluciona el producto a partir de los resultados completados de los bloques previos, añadiendo nuevos requerimientos y/o mejorando los que ya fueron completados, enfatizando siempre los requerimientos en función al valor que aportan al cliente.

A nivel básico, el modelo sigue el ciclo de vida del desarrollo incremental, observado en la figura 18, que a su vez está basado en el modelo cascada, incluyendo en cada en cada sub-fase etapa la posibilidad de revisar o mejorar funcionalidades desarrolladas previamente. El desarrollo incremental e iterativo no está orientado a un tipo específico de proyecto, sino a cierto tipo de cliente, siendo verdaderamente útil cuando el mismo necesita entregas rápidas, aunque sean parciales.

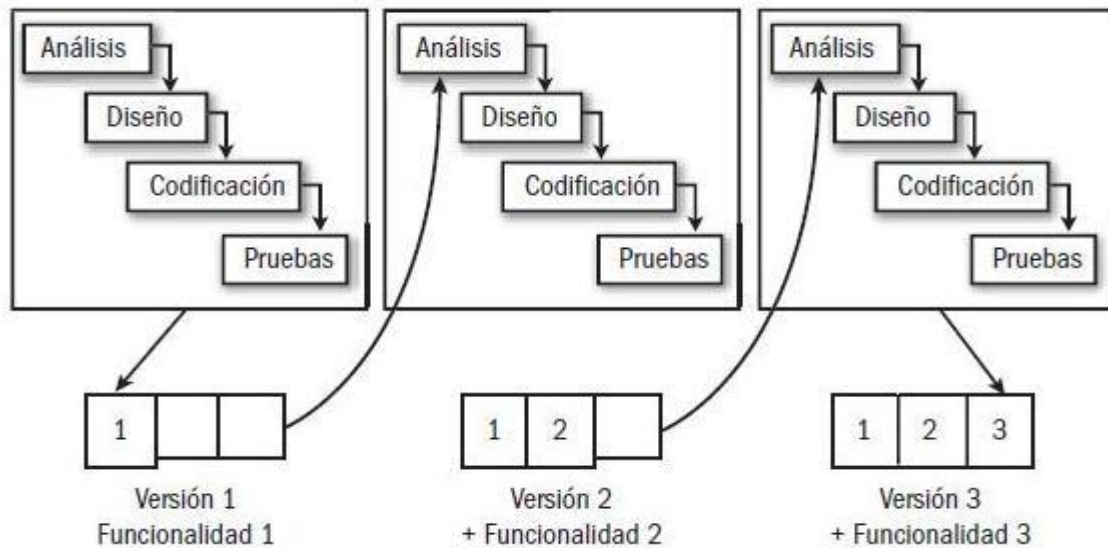


Figura 18: Ciclo de vida del desarrollo incremental (Cantone, 2006).

III.1.1 Etapas

- Análisis: esta fase involucra la captura de los requerimientos del sistema y su plasmado en un documento de especificación que sirve de guía para la siguiente fase.
- Diseño: esta fase envuelve entender los requerimientos del sistema y crear una idea general del producto final, definiendo adicionalmente la arquitectura del sistema.
- Implementación: se desarrollan los diferentes módulos de la aplicación, de acuerdo a los requerimientos y la arquitectura descrita en las fases previas.
- Pruebas: se prueba la funcionalidad de cada módulo desarrollado, verificando que cada módulo cumpla las especificaciones establecidas.

III.1.2 Ventajas

Entre las ventajas del Desarrollo Incremental e Iterativo se observan (Albaladejo, s/a):

- Se puede gestionar las expectativas del cliente de manera regular, pudiendo tomar decisiones en cada iteración, lo que resulta muy útil cuando el cliente no sabe exactamente que necesita o si se presenta la necesidad de realizar cambios a corto plazo.
- El cliente puede comenzar el proyecto con los requisitos de alto nivel que aportan mayor valor, incluso con requerimientos incompletos que pueden ser refinados en iteraciones sucesivas.
- El cliente puede obtener resultados importantes y usables desde las primeras iteraciones.

- Se puede gestionar de manera natural los cambios que van apareciendo durante el proyecto. La finalización de cada iteración es el lugar natural donde el cliente puede proporcionar sus observaciones tras examinar la versión producida y se pueden realizar cambios en la siguiente observación de acuerdo a las necesidades.
- Se minimiza el número de errores que se producen en el desarrollo y se aumenta la calidad del código, ya que cada iteración debe producir requisitos funcionales.

III.1.3 Desventajas

Entre las desventajas que se pueden observar del Desarrollo Incremental e Iterativo se encuentran (Albaladejo, s/a):

- La disponibilidad del cliente debe ser alta durante todo el proyecto, dada su participación continua tanto al inicio como al final de cada iteración.
- Cada iteración debe dar como resultado requisitos terminados, de tal forma que, de encontrar dificultades, puede involucrar retrasos en las entregas o la necesidad de tiempo extra de desarrollo.
- La relación con el cliente debe estar basada en los principios de colaboración más que tratarse de una relación contractual en la cual cada parte defiende únicamente su beneficio a corto plazo.
- Es necesario disponer de técnicas y herramientas que permitan hacer cambios fácilmente en el producto, de manera que pueda crecer en cada iteración de forma incremental sin realizar un gran esfuerzo adicional.
- Aumenta la dificultad de sistemas de alto tamaño en los cuales se prevea realizar múltiples integraciones entre las partes, ya que se diseñan de forma separada.

III.1.4 Artefactos

En base al Desarrollo Iterativo e Incremental se han creado varios *frameworks*, como el *Rational Unified Process* (RUP), que incluyen listas detalladas de artefactos a utilizar. Sin embargo, la metodología indica únicamente el uso de una lista de control de proyecto, para guiar el proceso de las iteraciones. La lista de control de proyecto contiene un historial de las tareas que requieren ser realizadas, incluyendo tanto nuevas funcionalidades a ser implementadas como áreas donde se debe mejorar el producto de iteraciones previas.

Dada la importancia en el *feedback* del usuario en el Desarrollo Incremental e Iterativo, es habitual utilizar algún artefacto que recoja formalmente los resultados de las interacciones del mismo con cada entrega realizada al final de una iteración. Adicionalmente, se recomienda el uso de cualquier artefacto de diseño, codificación o pruebas para apoyar la conclusión exitosa de dichas fases, utilizando en cada iteración aquellos artefactos que sean de utilidad y descartando los que no sean útiles, así se hayan usado en una o más iteraciones previas.

III. 1.5 Justificación

El Desarrollo Iterativo e Incremental está diseñado para trabajar con desarrollos con requerimientos poco definidos o altamente variables, por lo que es idóneo para la implementación de la solución planteada. Su flexibilidad ante las iteraciones, las fases y los artefactos lo hace indicado para lidiar con los requerimientos de la forma que sea de mayor utilidad para el cliente. Finalmente, la generación frecuente de entregas funcionales permitirá obtener *feedback* de forma regular de parte del cliente y algunos usuarios, para incorporar las observaciones de los mismos al proyecto de forma regular y rápida.

III.2 Metodología a seguir para el desarrollo de la solución

La metodología a utilizar para realizar la aplicación toma la estructura general del Desarrollo Incremental e Iterativo realizando incrementos pequeños, tal que el tiempo estimado para cada uno de los mismos se plantea entre uno y cinco días. En este período se realiza un análisis inicial del requerimiento a implementar, se diseña, se codifica el requerimiento y se realizan pruebas para verificar el comportamiento del módulo recién agregado, así como modificaciones necesarias para garantizar el nuevo requerimiento o para mejorar la calidad del producto general, producto de pruebas de fases anteriores o de cambios de diseño de la solución.

III.2.1 Artefactos

Como parte de la documentación de los principales productos de las distintas etapas de la metodología, se utilizan los siguientes artefactos:

III.2.1.1 Matriz de requerimientos funcionales

Documento en formato tabla donde se incluyen los requerimientos funcionales de la aplicación, incluyendo un ID, factibilidad, prioridad, estatus de completitud estimado y un orden en el cual se propone agregar cada funcionalidad. Esta matriz de requerimientos se actualiza semanalmente, y está dividida en dos secciones, requerimientos con interfaces asociadas y requerimientos del motor de la aplicación.

ID	Requerimiento Funcional de Aplicación	Web	Facebook	Fact Web	Fact FB	Prioridad
C.1	Persistencia de datos de usuario: el sistema debe almacenar el nombre, email y país del usuario	Si	No	Si	Si	Media
C.1.1	Datos opcionales del usuario: Fecha de nacimiento, sexo y Facebook ID.	Si	No	Si	Si	Media
C.2	Motor Backend: el sistema debe realizar respuestas asincronas al cliente para acelerar los reportes.	Si	Si	Si	Si	Media
C.3	Crawler de Likes: el sistema debe poder capturar los likes de los usuarios en FB para alimentar algoritmos de prediccion	No	Si	Si	Si	Media
C.3.1	El sistema debe almacenar los objetos a escanear	No	Si	Si	Si	Media
C.3.2	El sistema debe programar escaneos a los objetos periodicamente de forma configurable	No	Si	Si	Si	Media
C.3.3	El sistema debe permitir configurar el TTL de cada objeto a ser escaneado.	No	Si	Si	Si	Media
C.4	Motor Predicciones: el sistema debe generar ratings basados en los likes capturados para	Si	Si	Si	Si	Media
C.5	Administracion Reportes: el sistema debe almacenar los reportes y las acciones derivadas	Si	Si	Si	Si	Media
C.5.1	Versionamiento de frases: el sistema mantendra versiones previas de las frases cambiadas como resultado de un	Por Definir	Por Definir	Si	Si	Baja

Tabla 4: Matriz de requerimientos funcionales inicial de la aplicación.

III.2.1.2 Historias de usuario

Las historias de usuario son textos narrativos que describen una interacción entre el usuario y el sistema, enfocándose en la funcionalidad que el usuario obtendrá. Las historias del usuario se documentaron de forma física en papel o en la lista de tareas, siendo incorporadas a la matriz de requerimientos periódicamente, a medida que se definen los detalles de la implementación de la misma.

III.2.1.3 Diagrama de la base de datos

Diagrama de los objetos de la base de datos en formato entidad-relación extendido, actualizado al realizar cambios en la base de datos.

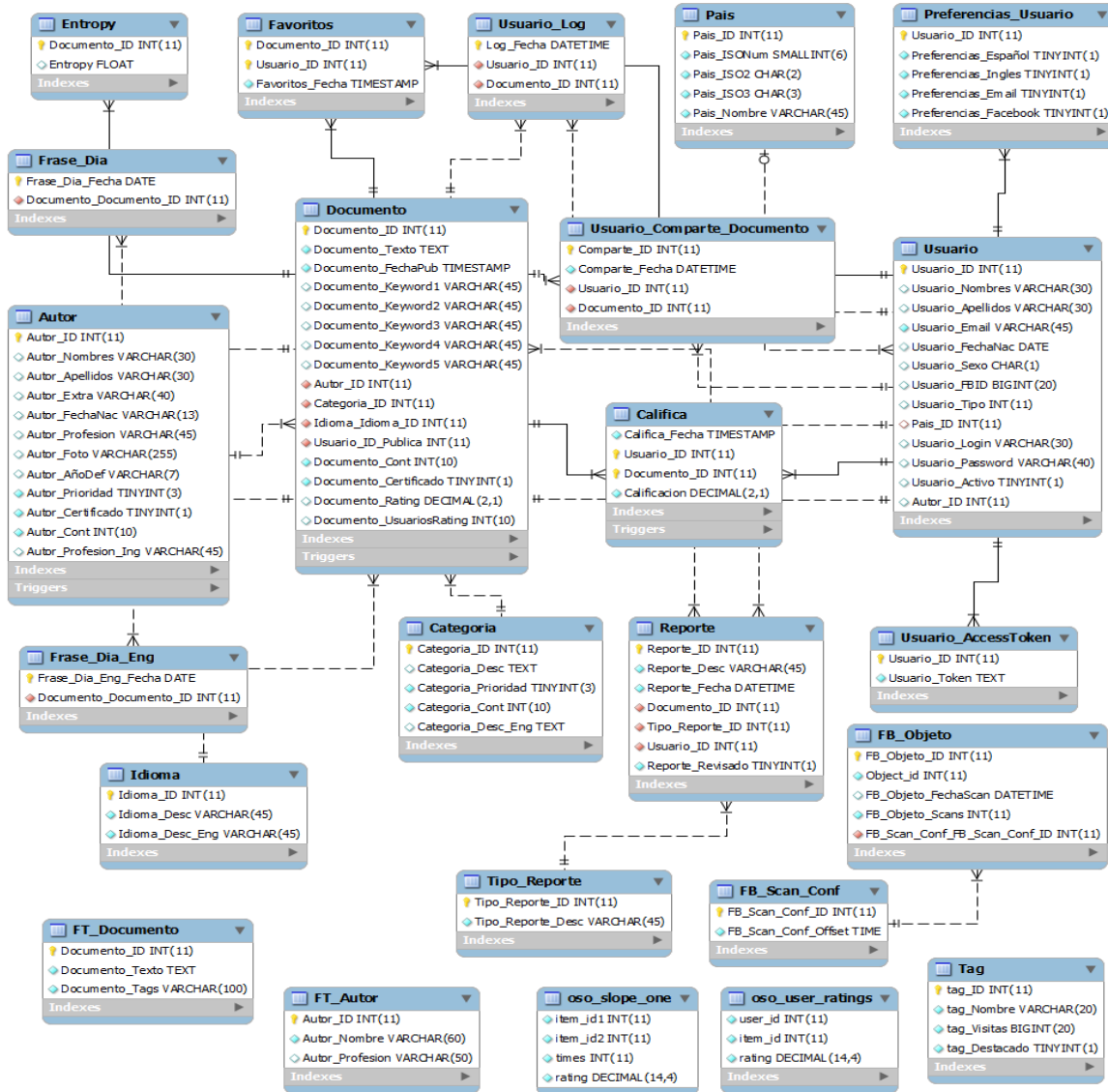


Figura 19: diagrama de entidad-relación extendido de la base de datos.

III.2.1.4 Lista de tareas

Una lista de las tareas planificadas en el incremento actual y de tareas propuestas para siguientes incrementos, siendo actualizada a medida que surjan los mismos en cada fase de análisis o diseño. La lista de tareas se alimenta principalmente de las historias de usuario, siendo en ocasiones contenedor de las mismas. Se mantuvo la lista en un pizarrón situado en un lugar de fácil acceso, para poder actualizarla o agregarle elementos rápidamente.

III.2.2 Justificación

La metodología propuesta permite realizar entregas constantes que facilitan el monitoreo del estado general del proyecto, permitiendo desarrollar inicialmente los componentes más visibles al usuario de la solución, lo que permite a su vez recopilar más datos de usabilidad de la misma y reunir un mayor conjunto de entrada para el motor de recomendación. Los artefactos utilizados fueron contruidos para documentar lo mínimo necesario a lo largo del desarrollo y permitir realizar cambios de forma rápida.

CAPÍTULO IV: DESARROLLO DE LA APLICACIÓN

Este capítulo ilustra el entorno de la aplicación, especificando sus objetivos generales y específicos, su alcance y la plataforma utilizada para su desarrollo e implementación. También presenta una descripción de la misma a mayor detalle, especificando los usuarios, los requerimientos, las principales funcionalidades aportadas, las interfaces principales, las historias de usuario más importantes y las pruebas realizadas.

IV.1 Objetivo General de la aplicación

Permitir al usuario consumir, publicar y compartir contenido literario relacionado a frases célebres.

IV.2 Objetivos Específicos de la aplicación

1. Presentar al usuario una interfaz atractiva que le permita interactuar con la base de datos de frases célebres.
2. Analizar el comportamiento de los usuarios.
3. Sugerir contenido dinámicamente de acuerdo al comportamiento registrado de los usuarios.
4. Incentivar la interacción del usuario mediante la capacidad de cambiar el contenido de la aplicación y la integración con plataformas de redes sociales.

IV.3 Alcance

La aplicación, titulada ***doKuot*** y ubicada en el dominio www.dokuot.com es un sitio web con características de red social orientada al contenido literario. La aplicación debe permitir a los usuarios formar una comunidad alrededor del contenido, por lo que se da un énfasis a la publicación y compartición del contenido generado en el sitio.

El sitio trata de transformar al usuario en un generador de contenido y, como tal, le permite introducir nuevas frases en el mismo, categorizarlas de acuerdo a unos parámetros muy generales, realizar calificaciones y comentarios y personalizar la forma en que los otros usuarios ven las contribuciones. Los contenidos introducidos por los administradores de la aplicación tienen una distinción visual asociada, y se prevé que un administrador puede realizar este proceso de “certificación” al contenido provisto por dichos usuarios.

De igual forma, se trata de ofrecer a cada usuario varias maneras de navegar a través de la gran cantidad de contenido ofrecido, bien sea mediante una búsqueda rápida, una búsqueda

parametrizada o un sistema de recomendaciones personalizadas de acuerdo a las calificaciones de los usuarios.

IV.4 Plataforma

El despliegue de la aplicación web se realizó bajo ambiente LAMP (Linux, Apache, MySQL, PHP) en un servidor remoto, ubicado en el huésped www.hostmonster.com. El servidor utilizado presenta las siguientes versiones:

- Apache 2.2.22
- PHP 5.3.16
- MySQL 5.1.65-community-log
- Kernel Linux: 2.6.32 x86

Adicionalmente, para el aspecto dinámico de las páginas web del lado del cliente, así como para las transferencias asíncronas, se utilizó *JavaScript* junto con las librerías *jQuery* versión 1.6.2 y *jQueryui* 1.8.23.

La aplicación fue desarrollada para uso en navegadores estándar (no móvil), teniendo en cuenta específicamente el caso de Internet Explorer, ofreciendo funcionalidades ligeramente reducidas para la versión 7, interfaces similares pero no idénticas en la versión 8 y una visualización correcta en la versión 9.

IV.5 Descripción de la aplicación

En esta sección se describen a detalle los diferentes elementos de la aplicación, así como sus usuarios, los requerimientos levantados, las funcionalidades implementadas, las principales interfaces e historias de usuarios y las pruebas realizadas junto con sus resultados.

IV.5.1 Usuarios

La aplicación está destinada a usuarios de Internet con un interés en el contenido descrito previamente. Entre estos usuarios se utilizaron tres perfiles de usuario a lo largo del diseño de la aplicación:

- Usuario consumidor: este usuario consume el contenido de la aplicación, pudiendo o no compartir el mismo, pero no realiza contribuciones de contenido. Muchas veces este usuario no se ha registrado en la aplicación.
- Usuario publicador: este usuario no solo consume contenido sino que también se interesa por contribuir contenido, primordialmente de autores famosos de su preferencia, hace comentarios y realiza calificaciones. Estas funcionalidades requieren el registro del usuario en la plataforma.

- Usuario escritor: este usuario disfruta tanto al consumir y publicar contenido de otros autores como de su autoría propia. Posee una página individual en la que se registran sus contribuciones. Se plantea en un futuro ofrecer la opción de “certificar” el contenido de este autor por parte de la aplicación.

Adicionalmente, se establece un rol de usuario administrador, que puede visualizar reportes y realizar cambios relativos al contenido y a los usuarios del sistema.

IV.5.2 Requerimientos

Los requerimientos identificados para la aplicación son los siguientes:

- Autenticación y registro de usuarios, utilizando una cuenta de Facebook o una cuenta de la aplicación.
- Consulta de ítems por autor, categoría, palabra clave (*tag*).
- Búsqueda de ítems específicos por autor, categoría, palabras clave, idioma, contenido del ítem y datos del autor, como su fecha de nacimiento o defunción o profesión.
- Compartir ítems con otros usuarios o miembros de las redes extendidas del usuario.
- Publicar ítems propios o de otros autores en la aplicación.
- Almacenar listas de ítems favoritos por usuario.
- Generar recomendaciones personalizadas por usuario.
- Reportar ítems por contenido erróneo.
- Administrar el contenido y los reportes enviados por los usuarios.

Similarmente, los requerimientos relacionados al sistema de recomendación son:

- Almacenar las calificaciones de los ítems hechas por los usuarios.
- Listar ítems recomendados en base a un ítem específico.
- Listar ítems recomendados en base a un usuario específico.
- Generar recomendación de un ítem a un usuario específico diariamente sin repetir ítems en un periodo de 30 días.

IV.5.3 Funcionalidades

La aplicación web, desde el punto de vista del usuario, es una experiencia de navegación continua. Sin embargo, el sistema se subdivide internamente en varios módulos, reutilizando una gran cantidad de código y reuniendo la mayor cantidad posible de código similar en el mismo archivo PHP. La estructura general del sistema es la siguiente:

- Autenticación y registro
 - Preferencias del usuario
- Consulta de contenido
 - Consulta por autores, categorías o palabras clave

- Favoritos
- Navegación
- Búsqueda de ítems
 - Búsqueda parametrizada
 - Búsqueda *FullText*
- Publicación de contenido
- Recomendación de contenido
 - Calificaciones
- Administración y reportes de ítems.

Los archivos de la aplicación utilizan una nomenclatura en la cual las páginas visibles al usuario final no contienen ningún prefijo, las paginas mostradas como *popups* o diálogos modales comienzan con el prefijo *pop_*, los scripts que procesan datos, generalmente de forma asíncrona, comienzan por el prefijo *fn_* y los archivos que forman parte de otras páginas y no pueden ser visualizados de forma individual (generalmente agrupando contenido igual o similar para muchas páginas) se nombran con el prefijo *inc_*. A continuación se presentan los módulos más importantes:

IV.5.3.1 Autenticación y registro

El módulo de autenticación y registro se encarga del registro de nuevos usuarios, la autenticación de los mismos en la plataforma y la integración con el proceso de autenticación de Facebook. La autenticación propia del sistema se realiza mediante la implementación de la librería de código abierto *phpUserClass* y las funciones de autenticación de Facebook se realizan mediante el uso del SDK de Facebook para PHP. Adicionalmente, se incluyen en este módulo las preferencias del usuario, donde se puede cambiar la contraseña, especificar ver ítems en español, inglés o ambos y recibir una frase recomendada por email o a través de Facebook diariamente. Los archivos relacionados con este módulo son:

- *Pop_login.php*: diálogo modal de inicio de sesión en el sistema.
- *Pop_registro.php*: diálogo modal para el registro de un nuevo usuario.
- *Pop_forgetpwd.php*: diálogo modal para recordar contraseña.
- *Fn_Verify.php*: contiene la lógica para verificar los datos de nombre de usuario y contraseña del usuario en la base de datos.
- *Fn_activate.php*: contiene la lógica para verificar que los datos del correo de activación enviado por el modulo sean correctos y activar la cuenta del usuario.
- *Fn_register.php*: contiene la lógica para almacenar los datos del registro del usuario en la base de datos.
- *Fn_loginFb.php*: contiene la lógica para interactuar con la plataforma de inicio de sesión de Facebook y almacenar los datos necesarios del usuario en la base de datos.
- *Fn_logout.php*: contiene la lógica para desconectar al usuario del sistema y/o de Facebook.

- Fn_preferencias.php: contiene la lógica para almacenar los cambios a la contraseña o a las preferencias del usuario.
- Preferencias.php: interfaz para visualizar y cambiar las preferencias del usuario.

IV.5.3.2 Consulta de contenido

El modulo de consulta de contenido se encarga de presentar al usuario diferentes ítems, de forma individual o agrupados por una característica común, como el autor, la categoría o una palabra clave. Similarmente se incluye en este módulo la consulta y selección de ítems favoritos, listas de ítems marcados por cada usuario. Las funciones para compartir el contenido del sitio en las redes sociales están incluidas en todas las interfaces de este modulo. Los archivos relacionados con este módulo son:

- Autor.php: Interfaz para visualizar los datos de un autor y los ítems de su autoría.
- Autores.php: Interfaz para visualizar la lista de autores paginados por letra.
- Categoria.php: Interfaz para visualizar los ítems que pertenecen a una categoría.
- Categorías.php: Interfaz para visualizar la lista de categorías de ítems.
- Frase.php: Interfaz para visualizar un ítem, calificarlo, realizar comentarios referentes al mismo o reportarlo a los administradores.
- Tag.php: Interfaz para visualizar los ítems que contienen una palabra clave en común.
- Favoritos.php: Interfaz para visualizar los ítems que el usuario ha agregado explícitamente a su lista de ítems favoritos.
- Fn_Favorito.php: contiene la lógica para agregar o eliminar un ítem de la lista de favoritos del usuario.
- Fn_VerMas.php: contiene la lógica para crear la lista de ítems de acuerdo al parámetro especificado (id del autor, id de la categoría, palabra clave o id del usuario para la lista de favoritos) y la paginación de la misma.
- Random_quote_daily.php: contiene la lógica para seleccionar una frase de forma aleatoria diariamente, ejecutándose como una tarea *cron*.

IV.5.3.3 Navegación

El módulo de navegación incluye principalmente la página principal, el menú del sitio y varios elementos que se ubican en la columna lateral derecha como atajos a las categorías o autores principales, así como a las palabras clave más populares. El módulo involucra los siguientes archivos:

- Daily_tags.php: script con la lógica para crear un XML con una lista de 25 palabras clave, incluyendo aquellas indicadas como prioritarias por los administradores y las más numerosas en la base de datos.
- fn_changelanguage.php: script con la lógica para permitir visualizar los ítems en un idioma específico por la duración de la sesión.

- Inc_about-main.php: interfaz diseñada para la columna lateral derecha con cuatro frases identificativas de la aplicación.
- Inc_autores-main.php: interfaz diseñada para la columna lateral derecha con algunos autores indicados como destacados en la base de datos.
- Inc_categorias-main.php: interfaz diseñada para la columna lateral derecha con algunas categorías indicadas como destacadas en la base de datos.
- Inc_frase-main.php: interfaz que muestra la frase del día en la página principal.
- Inc_frases.php: interfaz que muestra una lista de ítems que el usuario no ha calificado, ordenados por la entropía de las mismas. Si el usuario ha calificado más de 10 ítems, muestra ítems aleatorios en su lugar.
- Inc_header.php: interfaz de la cabecera de todas las paginas, incluyendo adicionalmente los enlaces del menú principal, un área dinámica de inicio de sesión / preferencias y la barra de búsqueda *FullText*.
- Inc_menu.php: funciones *JavaScript* para el funcionamiento del menú y algunos diálogos modales.
- Inc_nube-main.php: cargador de una animación flash ubicada en la columna lateral derecha con una nube animada de las palabras clave más comunes o destacadas.
- Index.php: pagina principal y punto de entrada esperado de la aplicación.

IV.5.3.4 Búsqueda

El módulo de búsqueda se encarga de la búsqueda de contenido de acuerdo a dos patrones de búsqueda: una búsqueda parametrizada de la base de datos donde el usuario puede ubicar ítems utilizando uno o más criterios de búsqueda y una búsqueda *Full-Text*, que trata de buscar autores o ítems basados en un *string* en lenguaje natural. Los archivos relacionados a este modulo son:

- Buscar.php: interfaz que permite al usuario realizar una consulta de ítems mediante un *string* de búsqueda y/o alguno de los siguientes parámetros: nombre, apellido, fecha de nacimiento, fecha de defunción o profesión del autor; categoría, idioma o palabras clave del ítem. Adicionalmente se le permite al usuario ordenar los resultados por el autor, la categoría, la fecha de publicación o la cantidad de visitas. Finalmente, se permite buscar solo ítems certificados o de autores certificados o ítems que no cumplen estas características.
- Resultados.php: vista donde se presenta al usuario los resultados de su búsqueda, conformado por una lista de los autores y/o ítems que coinciden con el *string* de búsqueda.
- Fn_buscar.php: script que contiene la lógica para realizar la búsqueda parametrizada generada por buscar.php.
- Fn_FullText.php: script que contiene la lógica para realizar la búsqueda en lenguaje natural. Esta búsqueda está basada en los índices *FullText* de las tablas *MyISAM* de *MYSQL* (Oracle Corporation, 2002), que interactúa con las tablas *FT_Autor* y

FT_Documento, réplicas de las tablas Autor y Documento con únicamente los campos necesarios para realizar los índices y permitir la funcionalidad de la búsqueda.

IV.5.3.5 Publicación

El módulo de publicación de contenido permite al usuario incluir nuevos ítems en la base de datos del sitio, bien sean originales o de autores conocidos. El sistema trata de identificar ítems repetidos, enviando un reporte a los administradores si un mismo ítem se trata de insertar con dos autores diferentes y rechazando un ítem repetido con el mismo autor existente. Teniendo en cuenta que el sistema puede dar cabida a SPAM y otros ítems inválidos, los nuevos ítems no están certificados hasta que un administrador los verifica y el sistema trata de identificar los ítems no validos y descartarlos, identificando el idioma del ítem, si contiene algún hipervínculo o si no proviene del dominio del sitio web. Los archivos relacionados a este módulo son:

- Fn_Autocomplete.php: archivo que contiene la lógica para generar una lista de 5 nombres de autores en base al string de búsqueda insertado, generado por publicar.php para tratar de autocompletar el campo del nombre del autor.
- Fn_Publicar.php: script que contiene la lógica para realizar la inserción de un nuevo ítem en la base de datos. Valida la ausencia de hipervínculos, el origen de la petición http, el idioma del texto del ítem y su posible existencia en la base de datos.
- Publicar.php: interfaz que permite al usuario ingresar un nuevo ítem en el sistema, especificar el autor o tomarlo de una lista que se autocompleta, agregar palabras clave que el usuario asocia al ítem y especificar una categoría para el mismo.

IV.5.3.6 Recomendaciones

El módulo de recomendaciones permite al usuario calificar los ítems del sistema y, mediante el almacenamiento de dichas calificaciones, genera recomendaciones de frases que pueden ser de interés al mismo. Siguiendo los pasos delineados por Lemire y McGrath (Lemire & McGrath, Implementing a Rating-Based Item-to-Item Recommender System in PHP/SQL, 2005), se realizó una implementación del algoritmo Slope One mediante PHP y MySQL, ubicando la gran parte de las operaciones en el motor de base de datos. La generación de las predicciones se realiza mediante las calificaciones en la tabla oso_user_ratings y la matriz de ítems a ítems en la tabla oso_slope_one. La matriz de ítems a ítems se actualiza antes de generar cada set de predicciones mediante el siguiente *procedure* de MySQL:

```
CREATE PROCEDURE `slope_one` ()
BEGIN
  DECLARE tmp_item_id INT ;
  DECLARE done INT DEFAULT 0 ;
  DECLARE mycursor CURSOR FOR
  SELECT DISTINCT
    item_id
  FROM
    oso_user_ratings ;
```

```

DECLARE CONTINUE HANDLER FOR NOT FOUND SET done = 1 ;

OPEN mycursor ;

WHILE
  (! done) DO FETCH mycursor INTO tmp_item_id ;

  IF (! done)
  THEN
  INSERT INTO oso_slope_one
  (SELECT
    a.item_id AS item_id1,
    b.item_id AS item_id2,
    COUNT(*) AS times,
    SUM(a.rating - b.rating) AS rating
  FROM
    oso_user_ratings a,
    oso_user_ratings b
  WHERE a.item_id = tmp_item_id
    AND b.item_id != a.item_id
    AND a.user_id = b.user_id
  GROUP BY a.item_id,
    b.item_id) ;

  END IF ;

END WHILE ;

CLOSE mycursor ;

END

```

La generación de predicciones para un usuario identificado por *\$userId* se realiza mediante la siguiente consulta:

```

SELECT
  s.item_id2
FROM
  oso_slope_one s,
  oso_user_ratings u
WHERE u.user_id = '$userId'
  AND s.item_id1 = u.item_id
  AND s.item_id2 != u.item_id
GROUP BY s.item_id2
ORDER BY SUM(u.rating * s.times - s.rating) / SUM(s.times) DESC
LIMIT 20

```

La generación de predicciones para ítems similares a un ítem identificado por \$itemId se realiza mediante la siguiente consulta:

```
SELECT
item_id2
FROM
oso_slope_one
WHERE item_id1 = '$itemId'
GROUP BY item_id2
ORDER BY SUM(rating / times)
LIMIT 20
```

Finalmente, los archivos relacionados con este modulo son:

- Recomendaciones.php: interfaz en la que se presenta al usuario los ítems recomendados por el sistema y le permite calificarlos.
- Fn_Rate.php: script que contiene la lógica para insertar una nueva calificación de un usuario a un ítem en la tabla Califica.
- Inc_recomendadas.php: script que contiene la lógica para generar las recomendaciones a presentar al usuario en recomendaciones.php.
- Update_ratings_daily.php: script que actualiza la calificación promedio de cada ítem en la tabla Documento en base a las calificaciones de los usuarios almacenadas en la tabla Califica diariamente.

IV.5.3.7 Administración y reportes.

El módulo de administración y reportes se subdivide en dos secciones interconectadas. El reporte de ítems permite a los usuarios enviar un reporte a los administradores del sitio, indicando un ítem con errores (ortográficos o gramaticales, autor incorrecto, duplicado) o inadecuado (consistente de SPAM o contenido inapropiado u ofensivo) para su revisión. La administración permite observar los reportes previamente descritos y realizar cambios de ser necesarios sobre la base de datos de ítems y de los usuarios, permitiendo al administrador visualizar los ítems, autores y usuarios ingresados al sistema, certificarlos, modificarlos o borrarlos; así como escoger las etiquetas “destacadas” a mostrar permanentemente en la interfaz. Los archivos públicos relacionados con este módulo son:

- Pop_reportar.php: interfaz que permite a un usuario reportar un ítem a los administradores, especificando la causa de una lista predefinida, y pudiendo especificar una explicación extendida.
- Fn_report.php: script que contiene la lógica para insertar el reporte enviado por el usuario en la base de datos.

Finalmente, los archivos privados a los que solo tiene acceso un administrador, ubicados en una subcarpeta con acceso restringido, son:

- `autores.php`: interfaz que presenta al administrador los autores incluidos en la base de datos en orden cronológico descendente, indicando el ID, nombre, ID de usuario (si aplica) y si tiene prioridad. Adicionalmente se indica si esta certificado y se permite certificarlo.
- `etiquetas.php`: interfaz que presenta al administrador las etiquetas asociadas a los ítems ordenando por su ID y si son destacadas, indicando adicionalmente el ID y el nombre y permitiéndole modificarlas, borrarlas o certificarlas.
- `frases.php`: interfaz que presenta al administrador las frases incluidas en la base de datos en orden cronológico descendente, incluyendo el ID, el texto de la frase, el autor y si son o no certificadas, permitiéndole modificarlas, borrarlas o certificarlas.
- `index.php`: interfaz que presenta la forma de inicio de sesión del módulo de administración.
- `reportes.php`: interfaz que presenta al administrador la lista de reportes ordenados por estatus y fecha, incluyendo el ID, la descripción, la fecha de reporte, la causa especificada, la frase indicada y el usuario que genero el reporte, permitiendo al administrador cambiar el estatus del reporte a revisado.
- `usuarios.php`: interfaz que presenta al administrador la lista de usuarios registrados en el sistema, especificando su ID, nombre, correo electrónico, fecha de nacimiento y sexo (si aplican), si el usuario se registro en el sitio web o mediante su cuenta de Facebook, si realizo la activación y su ID de autor (si lo posee). Se permite al administrador activar un usuario, modificar algunos de sus datos o borrarlo.
- `modautor.php`: interfaz para la modificación de los datos de un autor, permitiendo la modificación de los nombres, apellidos, fecha de nacimiento, año de defunción, profesión, foto, prioridad y ID del usuario asociado.
- `modfrase.php`: interfaz para la modificación de un ítem, permitiendo la modificación del texto, la categoría, palabras clave, ID de su autor, idioma y ID del usuario que la publicó.
- `modusuario.php`: interfaz para la modificación de un usuario, permitiendo la modificación de los nombres, apellidos, fecha de nacimiento, sexo y ID de autor asociado.
- `fn_Funciones.php`: script que contiene la lógica para certificar, borrar o modificar un ítem; certificar, borrar o modificar un autor; destacar, borrar o certificar una etiqueta; activar, modificar o borrar un usuario y modificar el estatus de un reporte.
- `fn_Verify.php`: script que contiene la lógica para verificar los datos de nombre de usuario y contraseña del usuario en la base de datos.

IV.5.4 Interfaces principales

El punto de entrada de la aplicación es la página principal, con la siguiente diagramación:

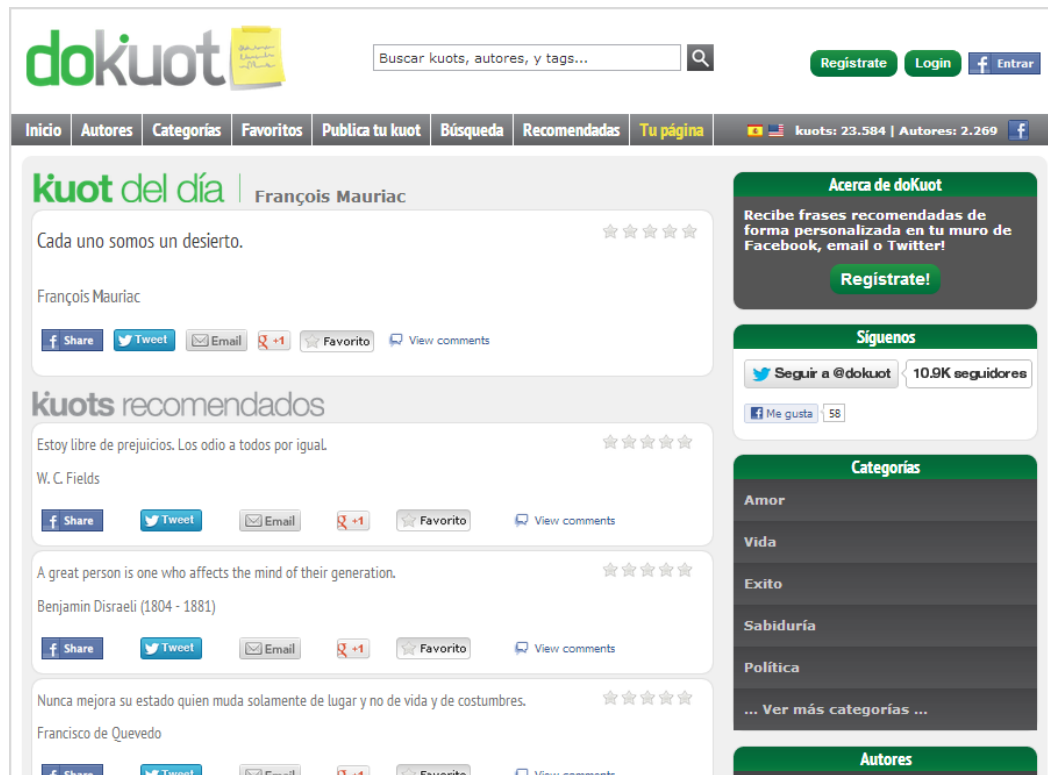


Figura 20: interfaz de la página principal.

El *login* o registro del usuario, mediante los dos botones ubicados en la esquina superior derecha, permite desplegar los siguientes diálogos modales:



Figura 21: Diálogo para autenticación de usuario.



Regístrate en doKout

Usuario:

Nombres:

Apellidos:

Email:

Password:

☒ Acepto los términos y condiciones

Enviar

O usa tu cuenta de...
 Facebook

Figura 22: Diálogo para el registro de un nuevo usuario.

Similarmente, dichos botones cambian la cabecera de las páginas de la siguiente manera, dependiendo del uso de una cuenta de usuario interna o una cuenta de Facebook, de la cual se puede extraer una foto:

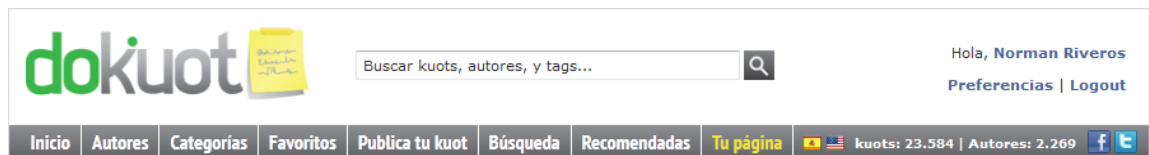


Figura 23: Vista de la cabecera para un usuario registrado en la aplicación.

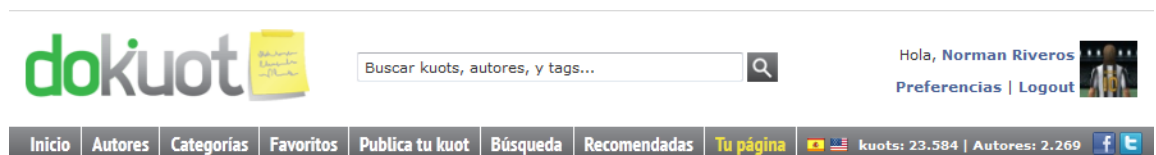


Figura 24: Vista de la cabecera para un usuario registrado mediante Facebook.

La vista de una frase, el ítem de la aplicación, incluye sus datos, los comentarios de los usuarios y las calificaciones de la siguiente manera:



Figura 25: Vista de una frase.

Los autores se listan mediante paginación por sus iniciales con la excepción de la lista inicial, que muestra los autores destacados.

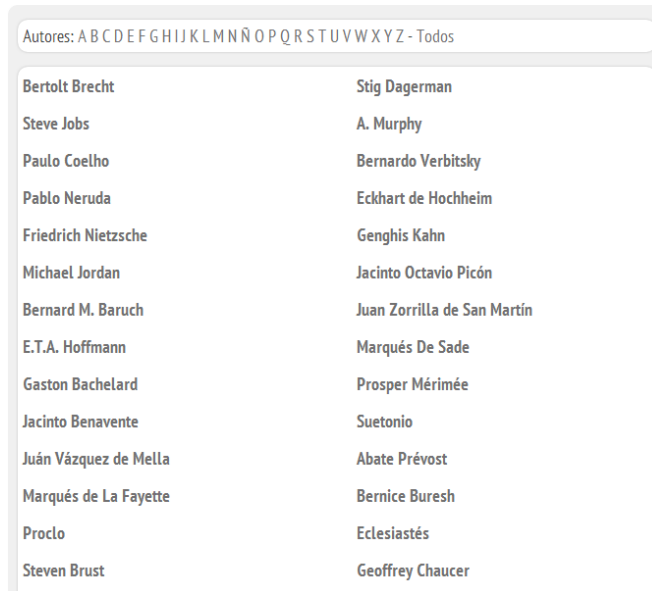


Figura 26: lista inicial de autores.

La interfaz para visualizar un autor específico, una vez seleccionado, incluye los datos del mismo y sus frases asociadas e indica si es certificado:

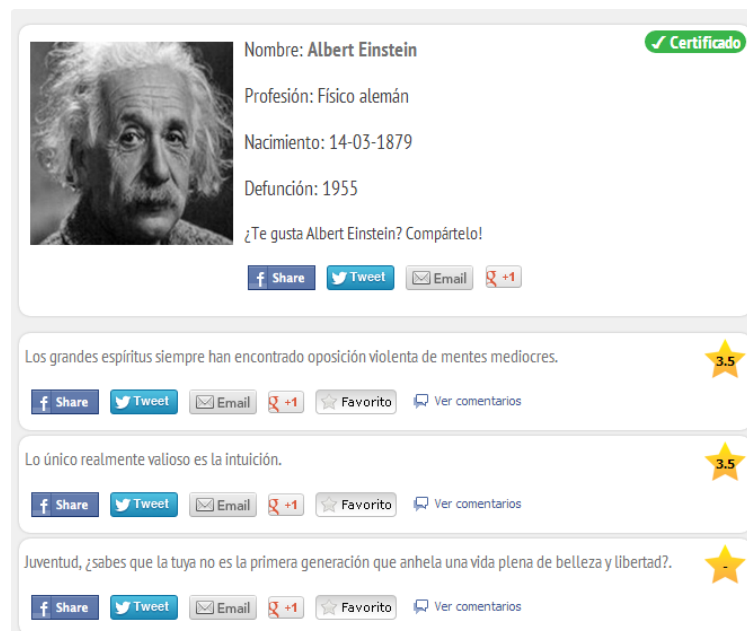


Figura 27: interfaz que permite visualizar un autor.

La lista de categorías se presenta al usuario de forma visual, ya que el número de categorías no es muy elevado, de la siguiente forma:



Figura 28: extracto de interfaz de lista de categorías.

Similarmente a la vista del autor, la interfaz que muestra una categoría seleccionada presenta una imagen asociada, el título de la categoría, la cantidad de ítems asociada a la misma y el resumen de los ítems, paginados por decenas.



Figura 29: interfaz de la categoría Éxito.

La interfaz para visualizar los favoritos tiene dos estados. Cuando el usuario no está conectado, le presenta un mensaje explicándole el concepto de frases favoritas e invitándolo a registrarse o conectarse. Cuando el usuario está conectado se presenta la lista de frases favoritas almacenadas en la base de datos para el mismo:



Figura 30: interfaz de la lista de frases favoritas del usuario.

La interfaz para publicar una frase se comporta de forma similar, mostrando la siguiente interfaz únicamente a los usuarios autenticados en la aplicación:

Figura 31: interfaz para publicar un nuevo ítem.

La página de búsqueda avanzada permite al usuario realizar una búsqueda de ítems mediante uno o más campos de los observados en la siguiente forma:

Búsqueda avanzada

Texto:

Autor:

Categoría:

Idioma:

Palabras clave:

Fecha de nacimiento del autor:

Profesión del autor:

Año de Defunción del autor: d.C.

Ordenar por:

☒ Buscar sólo Autores Certificados
☒ Buscar sólo Frases Certificadas

Figura 32: forma para búsqueda avanzada de frases.

Los resultados obtenidos se presentan de forma consistente con las interfaces previas que entregan al usuario una lista de frases. Se permite al usuario contraer o desplegar la forma de búsqueda para conducir una nueva consulta.

Búsqueda avanzada
Haz click aquí para ingresar otra búsqueda

Resultados (7)

Es delicioso dar con alguien que acepte las pequeñas ironías como expresiones de la mayor seriedad. ★
Aldous Huxley
[Share](#) [Tweet](#) [Email](#) [+1](#) [Favorito](#) [Ver comentarios](#)

La seriedad es sólo la corteza del árbol de la sabiduría; sin embargo, sirve para preservarla. ★
Confucio
[Share](#) [Tweet](#) [Email](#) [+1](#) [Favorito](#) [Ver comentarios](#)

La madurez del hombre es haber vuelto a encontrar la seriedad con la que jugaba cuando era niño. ★ 4.5
Friedrich Nietzsche
[Share](#) [Tweet](#) [Email](#) [+1](#) [Favorito](#) [Ver comentarios](#)

Cuando hables de Dios o de sus atributos, hazlo con toda seriedad y reverencia. ★
George Washington

Categorías
 Amor
 Vida
 Exito
 Sabiduría
 Política
 ... Ver más categorías ...

Autores
 Bertolt Brecht
 Steve Jobs
 Paulo Coelho
 Pablo Neruda
 Friedrich Nietzsche
 ... Ver más autores ...

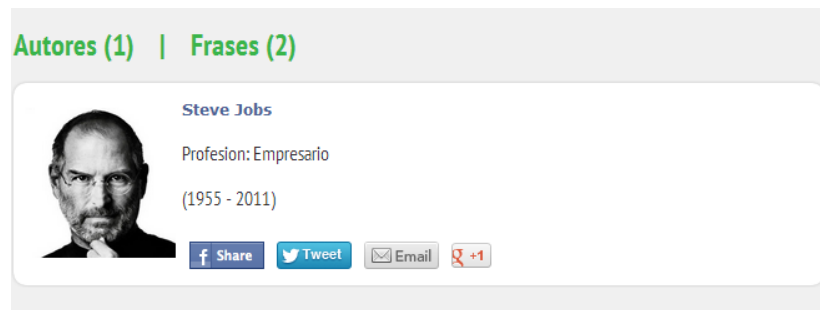
Nube de tags

Figura 33: resultados de búsqueda avanzada de frases.

La segunda modalidad de búsqueda se accede al utilizar el cuadro de búsqueda rápida ubicado en la cabecera, por ejemplo con la siguiente cadena de búsqueda:

Figura 34: recuadro de búsqueda *fulltext*.

Los resultados se presentan inicialmente con los autores (si existiesen como resultado de la búsqueda) y con acceso rápido a las frases, de la siguiente manera:

Figura 35: resultados de la búsqueda *fulltext*, vista de autores.Figura 36: resultados de la búsqueda *fulltext*, vista de frases.

La interfaz para ver recomendaciones en el sitio utiliza una vista muy similar a las listas de ítems, en la que se incluye un mensaje explicativo que el usuario puede ocultar y la calificación predicha por el sistema, de la siguiente manera:



Figura 37: interfaz que presenta al usuario una lista de ítems recomendados.

El módulo de reportes presenta al administrador una serie de interfaces listando las diferentes entidades de la aplicación. Los reportes que el usuario puede realizar acerca de un ítem, mediante el gráfico de la figura 39, son visualizados en la siguiente interfaz:

Frases	Autores	Etiquetas	Usuarios	Reportes				
Reportes ordenados por estatus y fecha								
	ID	Descripción	Fecha	Tipo	Frase	Usuario	Revisado	
☐	6	acento faltante en opción	2012-04-30 14:46:18	Errores ortográficos o gramaticales	1	5	☐	
☐	4	conocía esa frase dicha por Salvador Dalí	2012-04-30 14:43:32	Autor incorrecto	1	5	☐	
☐	3	refrán popular!	2012-04-30 14:42:41	Autor incorrecto	1	5	☒	
☐	2	mayúscula faltante luego de punto y seguido	2012-04-30 14:41:55	Errores ortográficos o gramaticales	1	5	☒	
☐	7	caracter incomprensible.	2012-05-07 09:44:19	Otro (por favor explique)	2265	5	☒	

Figura 38: interfaz del administrador para procesar los reportes.

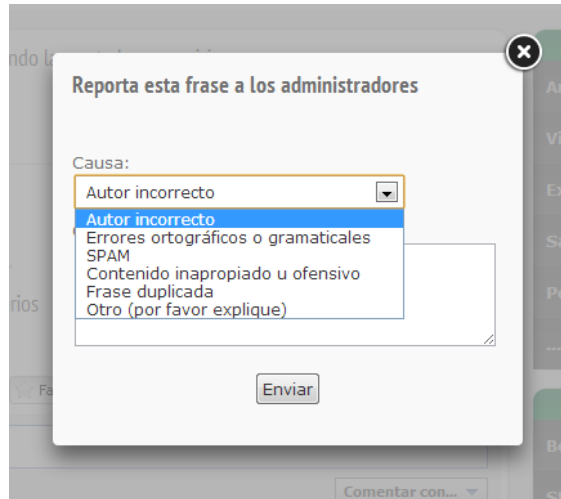


Figura 39: diálogo reportar frase.

Similarmente, el administrador puede observar una lista de las frases del sistema (figura 40) y modificar algunos de los datos de las mismas (figura 41) mediante las siguientes vistas:

Frases					
ID	Frase	Cert	Autor	Modificar	Borrar
☐ 25324	El poder es transitorio. Es muy distinto creerse el corazón de la patria que llevar la patria en el corazón.	☒	Henrique Capriles Radonski	Modificar	Borrar
☐ 25323	I'll be more enthusiastic about encouraging thinking outside the box when there's evidence of any thinking going on inside it.	☒	Terry Pratchett	Modificar	Borrar
☐ 25322	Metaphors have a way of holding the most truth in the least space.	☒	Orson Scott Card	Modificar	Borrar
☐ 25321	Entre mis posesiones mas preciadas están las palabras que no he pronunciado nunca.	☒	Orson Scott Card	Modificar	Borrar
☐ 25320	En el capitalismo el hombre es salvaje... En el comunismo es domesticado.	☐	Oliver Ortiz	Modificar	Borrar
☐ 25319	Pobreza es vivir en un lugar donde solo el dinero es riqueza.	☐	Oliver Ortiz	Modificar	Borrar
☐ 25318	El que persevera vende...	☐	Oliver Ortiz	Modificar	Borrar
☐ 25317	Una cosa es creerse el corazón de la Patria y otra cosa es llevar a la Patria en el corazón.	☒	Henrique Capriles Radonski	Modificar	Borrar
☐ 25316	The hottest places in hell are reserved for those who, in times of great moral crisis, maintain their neutrality.	☒	Dante Alighieri	Modificar	Borrar

Figura 40: lista de frases del sistema.

ID	25311
Texto	No llores porque ya se terminó... Sonríe porque sucedió.
Categoría	
Keyword1	
Keyword2	
Keyword3	
Keyword4	
Keyword5	
Autor ID	146
Idioma	1
Usuario	33
Guardar	

Figura 41: forma para modificar los datos de una frase.

Interfaces similares para los autores, las etiquetas y los usuarios se muestran en el anexo A.

IV.5.5 Historias de usuario principales

Las historias de usuario del sistema siguen el siguiente formato (Cohn, 2004):

Título: <título de la historia>

Descripción: *"Como un <rol> quisiera <acción> para <beneficio>"*.

Confirmaciones: <Criterios de aceptación de la historia>

Algunas de estas historias de usuario fueron posteriormente separadas en varias partes, documentando solo la descripción y sin incluir el <beneficio> para transformarlas en funcionalidades razonablemente simples de implementar en el tiempo. Algunas de las principales historias de usuario fueron las siguientes:

IV.5.5.1 Ver frases

Descripción: *"Como un usuario quisiera ver frases por alguno de varios parámetros en común para encontrar frases de mi agrado"*.

Confirmaciones:

- El usuario puede observar una lista de frases.
- Se despliega al usuario una lista de los posibles valores del parámetro seleccionado.
- La lista de frases corresponde a elementos del parámetro seleccionado.
- Si hay mas frases de las que caben en una página, el sistema provee la capacidad de ver la lista por páginas o secciones.

Esta historia de usuario de alto nivel fue separada en varias historias más pequeñas:

- “Como un usuario quisiera ver todas las frases de un autor”.
- “Como un usuario quisiera ver todas las frases relacionadas con un tópico”.
- “Como un usuario quisiera ver solo frases de un idioma específico”.
- “Como un usuario quisiera ver una lista de mis frases favoritas”.
- “Como un usuario quisiera ver una frase recomendada por el sistema”.

IV.5.5.2 Buscar frases

Descripción: *“Como un usuario quisiera buscar frases por uno o más parámetros para ubicar contenido específico de forma rápida”.*

Confirmaciones:

- El usuario puede observar una lista de resultados.
- Se presenta al usuario una lista de posibles valores para los parámetros seleccionados.
- El usuario puede ordenar la lista por el autor, la categoría, la fecha de publicación o la cantidad de visitas.
- Si hay más frases de las que caben en una página, el sistema provee la capacidad de ver la lista por páginas o secciones.

Esta historia de usuario fue dividida en las siguientes historias:

- “Como usuario quisiera buscar frases por su contenido”.
- “Como usuario quisiera buscar frases por su autor”.
- “Como usuario quisiera buscar frases por su categoría”.
- “Como usuario quisiera buscar frases por su idioma”.
- “Como usuario quisiera buscar frases por sus palabras clave”.
- “Como usuario quisiera buscar frases por la fecha de nacimiento del autor”.
- “Como usuario quisiera buscar frases por la fecha de defunción del autor”.
- “Como usuario quisiera buscar frases por la profesión del autor”.

IV.5.5.3 Publicar frase

Descripción: *“Como usuario quisiera publicar una nueva frase en la aplicación para comentarla y compartirla con mis amigos”.*

Confirmaciones:

- Para publicar una frase el usuario debe estar autenticado en la aplicación.
- El usuario especifica la categoría y hasta cinco palabras clave de la frase.

- El sistema autocompleta el nombre del autor a partir de las primeras letras ingresadas por el usuario.
- El sistema filtra las frases ya existentes en el sistema.
- El sistema reporta a un administrador los casos ambiguos.
- El sistema filtra las frases que contengan hipervínculos por considerarlas SPAM.
- El sistema filtra las frases que no estén en idiomas soportados por el sistema.

IV.5.5.4 Compartir contenido

Descripción: *“Como usuario quisiera poder compartir contenido para compartir mis intereses con mis amigos”.*

Confirmaciones:

- El usuario debe estar autenticado en la aplicación para poder compartir contenido.
- El usuario puede compartir contenido por correo electrónico, Twitter, Facebook y G+.

Esta historia de usuario de alto nivel fue separada en varias historias más pequeñas:

- “Como usuario quisiera compartir una frase”.
- “Como usuario quisiera compartir la página de un autor”.
- “Como usuario quisiera compartir las frases de una categoría”.
- “Como usuario quisiera compartir las frases con una palabra clave”.

IV.5.5.5 Reportar frase

Descripción: *“Como usuario quisiera poder reportar frases inadecuadas para poder remover ese contenido del sitio”.*

Confirmaciones:

- Para reportar una frase el usuario debe estar autenticado en la aplicación.
- Se presenta al usuario una lista de posibles causas del reporte.
- Se permite al usuario incluir una descripción extensiva del problema.

IV.5.5.6 Administración de frases

Descripción: *“Como administrador quisiera poder modificar las frases para poder corregir errores en las mismas”.*

Confirmaciones:

- El administrador ve una lista de frases.
- El administrador puede borrar una frase.

- El administrador puede certificar o descertificar una frase.
- El administrador puede modificar los campos de una frase exceptuando el ID.
- El autor especificado por el administrador debe existir.
- El usuario especificado por el administrador debe existir.

IV.5.5.7 Administración de usuarios

Descripción: *“Como administrador quisiera poder monitorear los usuarios para poder solventar problemas de los mismos”.*

Confirmaciones:

- El administrador ve una lista de usuarios.
- El administrador puede activar o desactivar un usuario.
- El administrador puede borrar un usuario.
- El administrador puede modificar los datos de un usuario, exceptuando el ID y el correo electrónico.

IV.5.6 Pruebas

Las pruebas de la aplicación web se realizaron mediante el cumplimiento de los criterios de aceptación de las historias de usuario definidas para la iteración en curso. Las interfaces se probaron utilizando los navegadores Chrome, Firefox e Internet Explorer versiones 7, 8 y 9. No se utilizaron pruebas automatizadas, sino que se realizaron pruebas manuales con usuarios de prueba, conformados por el equipo desarrollador en las etapas iniciales, y familiares y conocidos en etapas subsecuentes.

Las pruebas del sistema de recomendación se realizaron mediante un experimento en línea en dos fases. En una fase inicial, habiendo registrado en el sistema 473 calificaciones a 450 ítems diferentes con antelación, siete usuarios registraron una cuenta de usuario y se les presentó una lista de diez frases escogidas seleccionando los ítems de mayor entropía. Se les solicitó a los usuarios calificar dichas frases y posteriormente se les presentó una nueva lista de siete frases generadas por el sistema de recomendación. Finalmente, se les solicitó a los usuarios calificar estas frases y se almacenó la calificación del usuario, la calificación predicha por el sistema y el momento de la calificación. Dada la naturaleza del sistema, solo se generaron calificaciones positivas, por lo que se designó un verdadero positivo (VP) como una recomendación que predice una calificación mayor o igual a 4 y con un error menor a 0,5; por lo que los falsos positivos (FP) predicen una calificación mayor o igual a 4 y presentan un error mayor o igual a 0,5. De un total de 75 predicciones, se contaron 53 VP, 20 FP y 2 predicciones fuera del rango sin llegar a ser consideradas negativas. La precisión, calculada según la fórmula de la figura 7, fue de 0,726.

La media cuadrática del error (MCE) y la media absoluta del error (MAE), métricas de precisión ya descritas en la sección II.1.6.1.1, fueron calculadas según las formulas de las figuras 4 y 5 respectivamente con los datos obtenidos en esta fase de prueba, con valores obtenidos de 1,008454702 para la MCE y 0,721637028 para la MAE. De las 75 recomendaciones generadas, se observaron 27 recomendaciones únicas, con lo cual se estima una cobertura del 6%.

En la segunda fase de pruebas se incluyeron 21 usuarios, los 7 usuarios que participaron en la primera fase y 14 usuarios nuevos, utilizando el mismo procedimiento descrito para la fase 1. De un total de 178 predicciones, se contaron 105 VP, 54 FP y 19 predicciones fuera del rango sin llegar a ser consideradas negativas (valor mínimo predicho de 3,5). La precisión obtenida fue de 0,66, mientras que la MCE observada fue 0,61196227 y la MAE 0,6373933. De las 178 recomendaciones generadas se observaron 37 recomendaciones únicas, con lo cual se estima una cobertura del 7,02%.

En el gráfico de la figura 42 se puede observar el valor promedio de las predicciones hechas para un usuario junto con el promedio del valor absoluto del error luego de ambas fases de prueba, basado en el total de los datos:

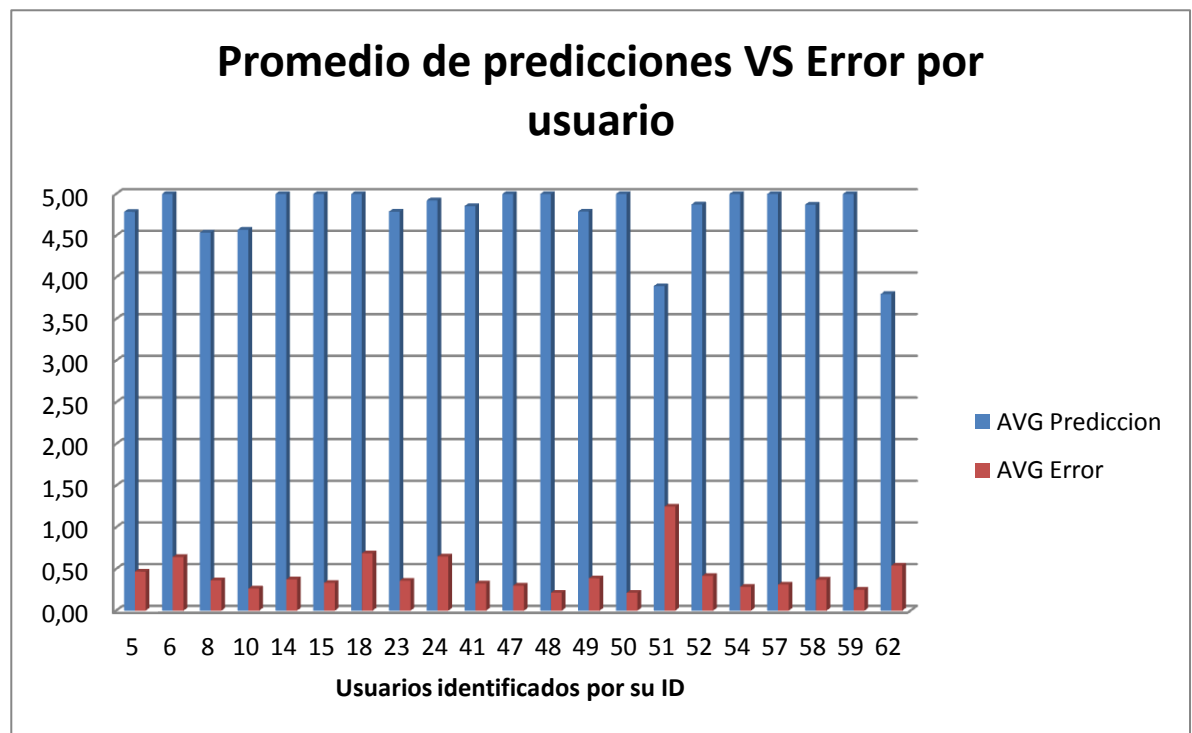


Figura 42: Gráfico de barras representativo del promedio de predicciones y el promedio del valor absoluto del error para cada usuario.

Como se observa en el grafico, únicamente el usuario 51 presento un error promedio mayor a 1. En el caso particular de este usuario, el sistema le presentó cuatro FP y dos VP en la

primera fase de pruebas. En la segunda fase, el SR presentó al usuario siete VP y cero FP, con valores predichos menores a la calificación del usuario en todos los casos.

La disminución de la MCE y la MAE de la fase uno a la fase dos provee una tendencia, acorde con la teoría de los sistemas de filtrado colaborativo, a la disminución del error a medida que aumenta el número de usuarios activos y de calificaciones registradas. Se atribuye la disminución de la precisión a la existencia de recomendaciones realizadas a los usuarios que, a pesar de ser calificadas por los usuarios de forma positiva, son incluidas como FP por tener un error mayor a 0,5. La tendencia de estos valores entre las dos fases de pruebas se pueden observar en la figura 43:

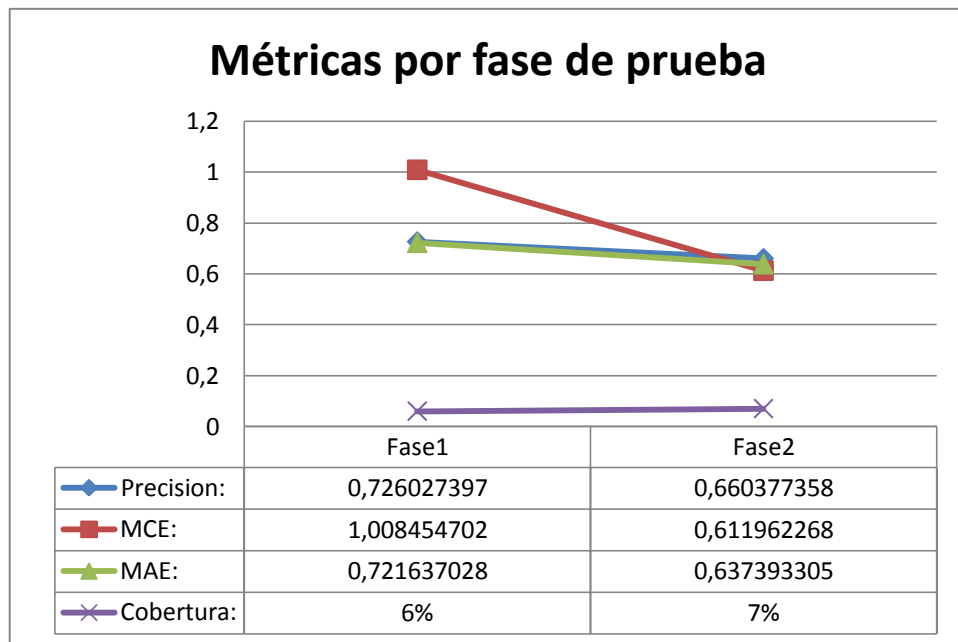


Figura 43: Valores de las métricas Precisión, MCE, MAE y Cobertura en las fases de prueba.

RESULTADOS DEL TRABAJO ESPECIAL DE GRADO

Como resultado principal del Trabajo Especial de Grado se obtuvo una versión apta para el lanzamiento del portal web junto con un prototipo del sistema de recomendación, implementando el algoritmo *Open Slope One*.

Otro resultado lo arrojan los datos extraídos de las pruebas realizadas, los cuales indican que la efectividad del motor de predicciones aumentará a lo largo del tiempo, a medida que más usuarios se incorporen a la plataforma. Se puede observar más información recopilada como resultado de las pruebas en el anexo B.

Se desarrolló una aplicación web que permite la integración entre el contenido y las redes sociales Facebook, Twitter y G+, observándose una complejidad relativamente baja para interactuar con Twitter y G+ y una más alta con Facebook, causada especialmente por la baja calidad de la documentación publica disponible y de los frecuentes cambios realizados en esta aplicación social durante el desarrollo del TEG.

Similarmente, se desarrolló un módulo de administración que permite interactuar con varios elementos de la base de datos. Si bien este módulo cuenta con interfaces de usuario menos logradas, ya que no van a ser visualizadas por el usuario general, provee al administrador las funcionalidades necesarias para manejar el contenido del sitio y algunos aspectos de su visualización sin necesidad de modificar el código.

Finalmente, se elaboraron documentos para el soporte y la extensión del sitio en futuras oportunidades, como la matriz de requerimientos funcionales, que incluye requerimientos identificados que aun no han sido desarrollados o implementados, el diagrama de entidad relación extendido de la base de datos, el documento de historias de usuario y los suficientes comentarios en el código fuente de los scripts de generación de contenido para explicar el flujo de ejecución del mismo.

CONCLUSIONES Y RECOMENDACIONES

Los requerimientos del sistema fueron capturados y modificados constantemente a lo largo del proyecto, quedando plasmados en la matriz de requerimientos funcionales y en las historias de usuario. La base de datos del sistema evolucionó a causa de estos cambios, partiendo de un estado inicial con alrededor de diez tablas al estado plasmado en este documento con veinticuatro tablas. El sitio web fue lanzado al público luego de varias iteraciones que tomaron alrededor de cuatro meses de trabajo, incluyendo muchas de las características de interacción con redes sociales desde el lanzamiento.

El sistema de recomendación fue implementado siguiendo los lineamientos descritos por uno de sus creadores y adaptándolos al problema en particular de este trabajo de investigación. Las pruebas del mismo se realizaron mediante dos experimentos en línea en los que se evaluaron algunas métricas de precisión y de cobertura.

La metodología utilizada permitió reaccionar rápidamente a los cambios, que ocurrieron frecuentemente, permitiendo desarrollar primero aquellas funcionalidades que aportaron mayor valor al cliente a lo largo del desarrollo. Similarmente, permitió realizar implementaciones iniciales rápidas de funcionalidades que luego fueron descartadas o reubicadas, sin conllevar un mayor retraso en el proyecto a gran escala.

El presente Trabajo Especial de Grado presenta otro caso de uso exitoso para el algoritmo Slope One en sistemas de recomendación de bajo perfil, permitiendo un desarrollo veloz y de bajo costo que permite realizar actualizaciones en caliente y presentar recomendaciones de forma rápida en servidores compartidos. El proceso descrito puede ser aplicado a otros problemas de características similares para generar recomendaciones y medir la efectividad del sistema de recomendación de forma somera.

Futuros trabajos pueden comenzar por realizar una nueva medición para comprobar si las tendencias observadas en las pruebas se cumplen a lo largo del tiempo. Similarmente se puede realizar una comparación de la implementación actual con otros algoritmos o esquemas de recomendación utilizando las métricas descritas.

Otra opción para una mejora del sistema ante el problema del arranque en frío, esta vez de ítems y no del sistema completo, sería implementar robots de calificaciones, de tal forma que un nuevo ítem pueda ser recomendado con mayor frecuencia ya que posee calificaciones generadas por el sistema de acuerdo a ciertos parámetros; posiblemente continuando el trabajo propuesto por Park, et al (Park, Pennock, Madani, Good, & DeCoste, 2006). Esta mejora no fue implementada como parte de este trabajo por estar fuera del alcance del mismo.

REFERENCIAS BIBLIOGRÁFICAS Y DÍGITALES

- Albaladejo, X. (s/a). *Desarrollo iterativo e incremental*. Recuperado el 1 de Febrero de 2012, de Proyectos Agiles: <http://www.proyectosagiles.org/desarrollo-iterativo-incremental>
- Boyd, D. (2011). Social Network Sites as Networked Publics: Affordances, Dynamics, and Implications . En Z. Papacharissi, *A Networked Self* (págs. 39-58). New York: Taylor & Francis.
- Boyd, D., & Ellison, N. (2007). Social Network Sites: Definition, History, and Scholarship. *Journal of Computer-Mediated Communication*.
- Burke, R., & Ramezani, M. (2010). Matching Recommendation Technologies and Domains. En F. Ricci, L. Rokach, B. Shapira, & P. Kantor, *Recommender Systems Handbook* (págs. 367-386).
- Cantone, D. (2006). Ciclo de vida del software. En D. Cantone, *Implementacion y Debugging* (págs. 16-35). MP EDICIONES.
- Desrosiers, C., & Karypis, G. (2011). A comprehensive survey of neighborhood-based recommendation methods. En F. Ricci, L. Rokach, B. Shapira, & P. Kantor, *Recommender Systems Handbook* (págs. 107-144). Springer. Obtenido de Karypis Lab: <http://glaros.dtc.umn.edu/gkhome/node/748>
- Groh, G., & Ehmig, C. (2007). Recommendations in Taste Related Domains: Collaborative Filtering vs. Social Filtering. *GROUP '07: Proceedings of the 2007 international ACM conference on Supporting group work* (págs. 1-8). New York: ACM.
- Haw, K., & Watermillier, A. (2010). *Comparing Content Filtering and Collaborative Filtering in Recommender Systems: A Problem Analysis*. Recuperado el 11 de 1 de 2012, de University of Northern Iowa: <http://www.cs.uni.edu/~schafer/courses/previous/161/Fall2010/proceedings/papers/paperF.pdf>
- Herlocker, J., Konstan, J., Borchers, A., & Riedl, J. (Agosto de 1999). *GroupLens Research / Publications*. Recuperado el 19 de Enero de 2012, de GroupLens Research - Department of Computer Science and Engineering at the University of Minnesota: <http://www.grouplens.org/node/229>
- Herlocker, J., Konstan, J., Terveen, L., & Riedl, J. (2004). Evaluating Collaborative Filtering Recommender Systems. *ACM Transactions on Information Systems* (págs. 5-23). ACM.

- Lemire, D., & Maclachlan, A. (2005). Slope One Predictors for Online Rating-Based Collaborative Filtering. *SIAM Data Mining (SDM 05)*, (págs. 21-23). Newport Beach, California, USA.
- Lemire, D., & McGrath, S. (Enero de 2005). *Implementing a Rating-Based Item-to-Item Recommender System in PHP/SQL*. Recuperado el 21 de Enero de 2012, de Daniel Lemire's Blog: <http://lemire.me/fr/abstracts/TRD01.html>
- Lops, P., de Gemmis, M., & Semeraro, G. (2010). Content-based Recommender Systems: State of the Art and Trends. En F. Ricci, L. Rokach, B. Shapira, & P. Kantor, *Recommender Systems Handbook* (págs. 73-105).
- Massa, P., & Avesani, P. (2004). Trust-aware collaborative filtering for recommender systems. *Proceedings of the International Conference on Cooperative Information Systems, CoopIS*, (págs. 492-508). Larnaca.
- Melville, P., & Sindhvani, V. (2010). Recommender Systems. En C. Sammut, & G. I. Webb, *Encyclopaedia of Machine Learning* (págs. 829-837).
- Oracle Corporation. (2002). *Full-Text Search Functions*. Recuperado el Octubre de 2012, de MySQL 5.0 Reference Manual: <http://dev.mysql.com/doc/refman/5.0/en/fulltext-search.html>
- Ricci, F., Rokach, L., & Shapira, B. (2010). Introduction to Recommender Systems Handbook. En F. Ricci, L. Rokach, B. Shapira, & P. Kantor, *Recommender Systems Handbook* (págs. 1-29). Springer.
- Sarwar, B., Karypis, G., Konstan, G., & Reidl, J. (2001). Item-based collaborative filtering recommendation algorithms. *WWW '01 Proceedings of the 10th international conference on World Wide Web* (págs. 285 - 295). New York: ACM New York.
- Shani, G., & Gunawardana, A. (2010). Evaluating Recommendation Systems. En F. Ricci, L. Rokach, B. Shapira, & P. Kantor, *Recommender Systems Handbook* (págs. 257-297). Springer.

ANEXO A – INTERFACES DEL MÓDULO DE ADMINISTRACIÓN

A continuación se presentan otras interfaces pertenecientes al módulo administrativo:

Autores ordenados cronológicamente							
	ID	Nombre	Usuario	Prioridad	Certificado	Modificar	Borrar
☐	2285	Terry Pratchett			☐	Modificar	Borrar
☐	2282	Orson Scott Card			☐	Modificar	Borrar
☐	2281	Oliver Ortiz	44		☐	Modificar	Borrar
☐	2280	Henrique Capriles Radonski			☐	Modificar	Borrar
☐	2279	Blair Waldorf			☐	Modificar	Borrar
☐	2277	Un monje en la Inia			☐	Modificar	Borrar
☐	2275	William Wallace			☐	Modificar	Borrar
☐	2274	Albert Einstein			☐	Modificar	Borrar
☐	2273	Ana Sarahi Castillo Castor	33		☐	Modificar	Borrar
☐	2272	Lair Ribeiro			☐	Modificar	Borrar
☐	2270	Alex The Brit guy			☐	Modificar	Borrar
☐	2269	Isabel Ibañez	20		☒	Modificar	Borrar
☐	2267	William Gibson			☒	Modificar	Borrar

Figura 44: lista de autores del sistema.

ID	2285
Nombres	Terry
Apellidos	Pratchett
Extra	
Fecha de Nacimiento(YYYY-MM-DD)	
Año de Defuncion	
Profesion	
Foto	
Prioridad	99
Usuario	
	Guardar

Figura 45: forma para modificar los datos de un autor.

Etiquetas ordenadas por Destacado & ID					
	ID	Nombre	Destacado	Modificar	Borrar
☐	247	amor	☒	Guardar Cambios	Borrar
☐	251	teamwork	☒	Guardar Cambios	Borrar
☐	1	compañía	☐	Guardar Cambios	Borrar
☐	2	compasion	☐	Guardar Cambios	Borrar
☐	3	comunismo	☐	Guardar Cambios	Borrar
☐	4	confianza	☐	Guardar Cambios	Borrar

Figura 46: lista de etiquetas del sistema.

Frases ordenadas por Destacado									
	ID	Nombre	Email	Fecha de Nacimiento	Sexo	FB / L	Activo	Autor ID	Modificar
☐	NN	Anibal *****	*****@hotmail.com	1968-**-**	m	FB	☒		Modificar
☐	63	Jose Antonio ***** (Jota)	*****@gmail.com	0000-00-00		L	☒		Modificar
☐	62	Luis *****	*****@yahoo.com	1978-**-**	m	FB	☒		Modificar
☐	61	Ilsesita *****	*****@hotmail.com	1987-**-**	f	FB	☒		Modificar

Figura 47: lista de usuarios del sistema (algunos datos ocultos mediante asteriscos por privacidad)

ID	5
Nombres	Norman
Apellidos	Riveros
Email	norman.riveros@gmail.com
Fecha de Nacimiento (YYYY-MM-DD)	1981-11-15
Sexo (f / m)	m
Autor ID	2264
	Guardar

Figura 48: forma para modificar los datos de un usuario.

ANEXO B – RESULTADOS DE LAS PRUEBAS

A continuación se presentan diferentes gráficos obtenidos de los resultados obtenidos en las pruebas de la aplicación.

Los gráficos de las figuras 49, 50 y 51 muestran el porcentaje de verdaderos y falsos positivos y las predicciones fuera de rango en las fases de prueba 1 y 2 y del total de las predicciones. Los gráficos muestran como el porcentaje de VP baja de la primera fase de prueba a la segunda, manteniéndose en general alrededor de un 63%, un valor adecuado para un sistema con una cantidad relativamente baja de calificaciones.

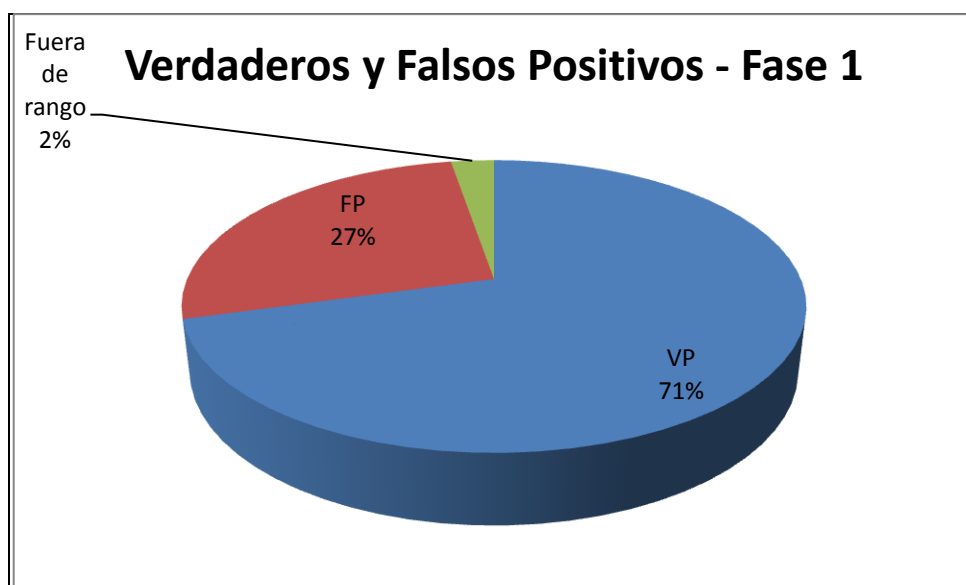


Figura 49: Gráfico de tortas del porcentaje de verdaderos y falsos positivos en la primera fase de prueba.

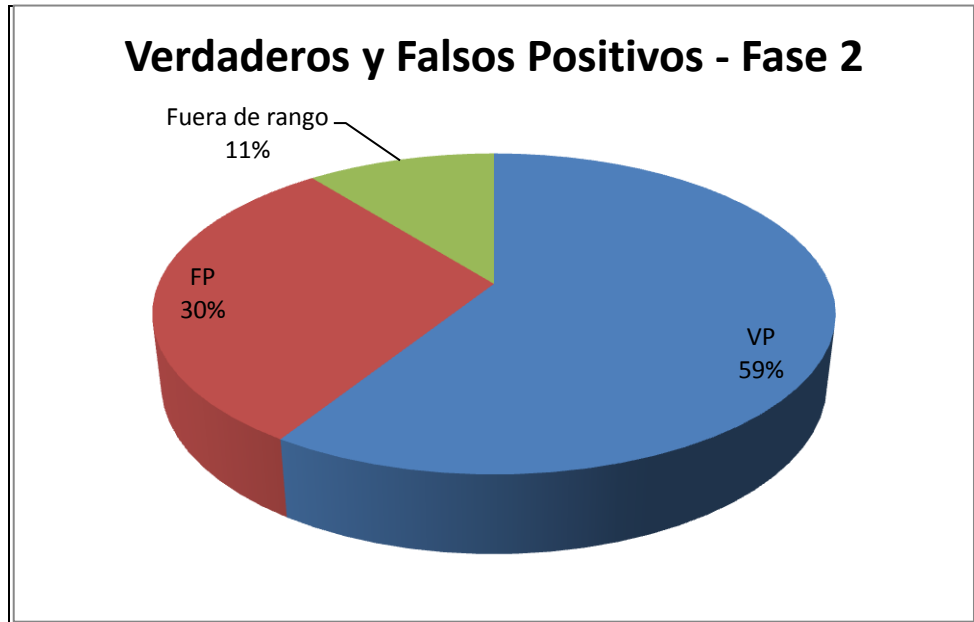


Figura 50: Gráfico de tortas del porcentaje de verdaderos y falsos positivos en la segunda fase de prueba.

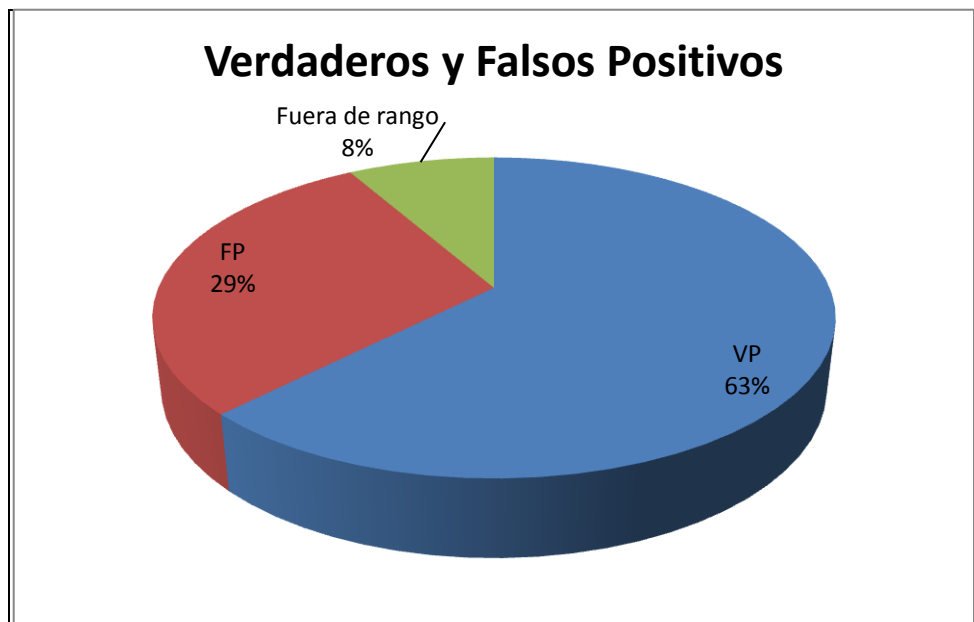


Figura 51: Gráfico de tortas del porcentaje de verdaderos y falsos positivos en la fase de pruebas.

El gráfico de la figura 52 toma los datos de los usuarios que participaron en ambas fases de prueba y muestra la variación del error promedio para cada usuario entre las mismas. Se observa claramente como un único usuario, de id 58, observó un aumento en el error promedio de las frases recomendadas por el sistema.

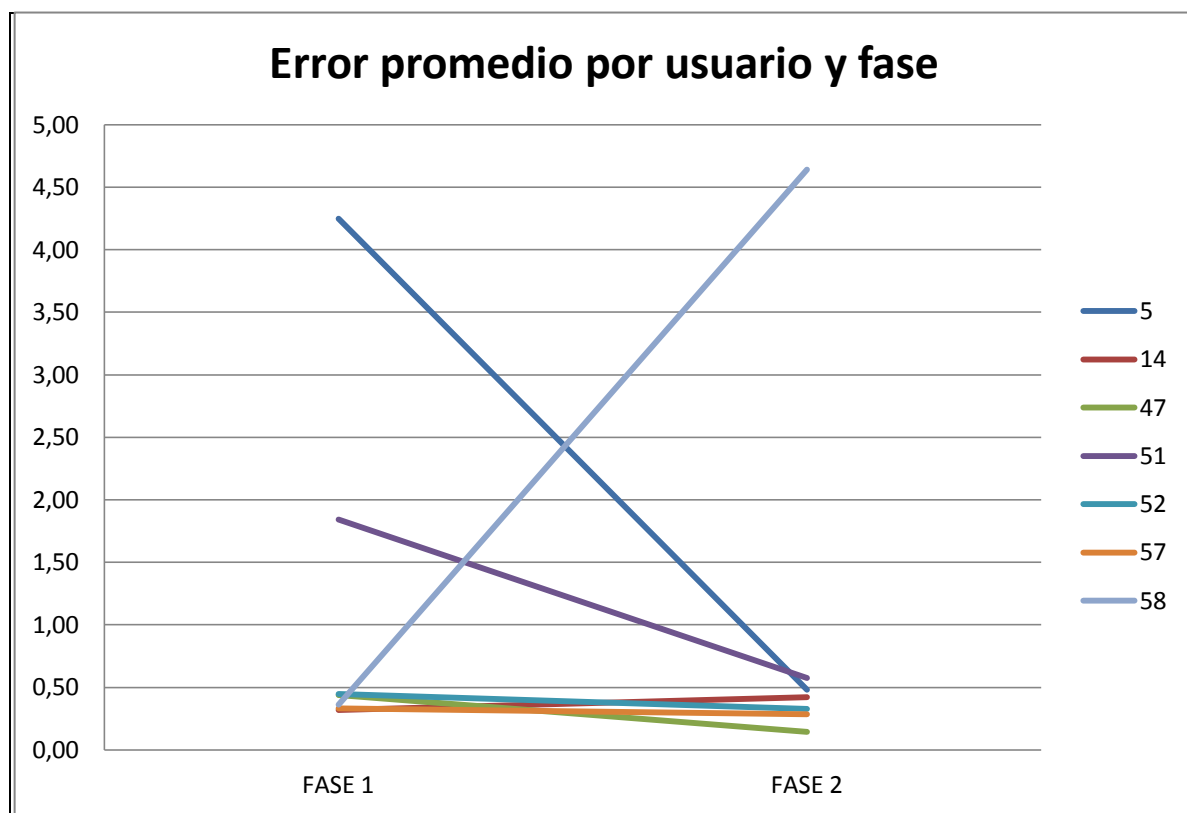


Figura 52: Error promedio para los usuarios participantes en las 2 fases, separado por fase.

Finalmente, las tablas 5, 6 y 7 muestran un resumen de los valores obtenidos para las fases de prueba 1 y 2 y para el total de las calificaciones. Cada tabla incluye el número de calificaciones existentes en el sistema antes de comenzar cada fase, o el total de calificaciones en el resumen final, el total de calificaciones registradas en la fase y los valores de verdaderos y falsos positivos, la precisión (ver figura 7), el recall y la tasa de FP que siempre toman un valor 1 ya que el sistema no realiza recomendaciones negativas, la media cuadrática del error (ver figura 4) y la media absoluta del error (ver figura 5) y la cobertura del catálogo, calculada por la relación entre los diferentes ítems recomendados y los ítems que podían ser recomendados (aquellos con al menos una calificación). La precisión promedio del sistema se encuentra alrededor del 72% y se observa como las métricas de error disminuyen mientras que la cobertura del catálogo aumenta a lo largo del tiempo.

Calificaciones previas:	473
Total:	75
VP	53
FP	20
Precisión:	0,726027397
Recall:	1

Tasa de FP:	1
MCE:	1,008454702
MAE:	0,721637028
Cobertura:	6%

Tabla 5: Tabla resumen de los resultados de la primera fase de pruebas.

Calificaciones previas:	676
Total:	178
VP	105
FP	54
Precisión:	0,660377358
Recall:	1
Tasa de FP:	1
MCE:	0,611962268
MAE:	0,637393305
Cobertura:	7%

Tabla 6: Tabla resumen de los resultados de la segunda fase de pruebas.

Calificaciones:	253
VP	159
FP	61
Precisión:	0,72272727
Recall:	1
Tasa de FP:	1
MCE:	0,75163637
MAE:	0,66348314
Cobertura:	8,73%

Tabla 7: Tabla resumen de los resultados de ambas fases de prueba.